



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE  
TELECOMUNICACIÓN

Titulación :

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

“MÓDULO DE LOCALIZACIÓN E INSERCIÓN AUTOMÁTICA  
DE REFERENCIAS EXTERNAS ASOCIADAS AL CÓDIGO  
PUBLICADO EN WEBS DE BUENAS PRÁCTICAS DE  
DESARROLLO”

Gabriel Rocamador Murillo

Jesús Villadangos Alonso

Pamplona, Junio 2010

## ÍNDICE

1. Introducción	4
2. Arquitectura del sistema	6
3. Herramientas y pruebas de concepto	10
3.1 WorkFlow	10
3.2 Búsquedas	12
3.2.1 Consultas Html	13
3.2.2 Consultas Xml	15
4. Metodología de desarrollo aplicada: Scrum	19
4.1 Metodologías Ágiles	19
4.2 Scrum	20
4.3 Scrum aplicado al proyecto	22
5. Desarrollo: Sprint 1	24
5.1 Selección de requisitos	24
5.2 Seguimiento de las tareas	25
5.3 Diagramas de clases y listas creadas	27
5.4 Incidencias destacables	29
5.5 Producto entregable	30
6. Desarrollo: Sprint 2	33
6.1 Selección de requisitos	33
6.2 Seguimiento de las tareas	34
6.3 Diagramas de clases y listas creadas	36
6.4 Incidencias destacables	37
6.5 Producto entregable	37
7. Desarrollo: Sprint 3	41
7.1 Selección de requisitos	41
7.2 Seguimiento de las tareas	42
7.3 Diagramas de clases y listas creadas	43
7.4 Incidencias destacables	44
7.5 Producto entregable	45

8. Desarrollo: Sprint 4	49
8.1 Selección de requisitos	49
8.2 Seguimiento de las tareas	50
8.3 Diagramas de clases y listas creadas	51
8.4 Incidencias destacables	52
8.5 Producto entregable	53
9. Conclusiones y líneas futuras	55
9.1 Conclusiones	55
9.2 Líneas Futuras	56
10. Bibliografía	57

## 1. Introducción

La creación de software no es una tarea sencilla, disponemos de una gran multitud de lenguajes, y la tecnología avanza a un ritmo exponencial. La evolución de la web obliga al desarrollador a adaptarse continuamente, ésto se consigue de manera mas sencilla a través de una puesta en común tanto de conocimientos como de código. Así han surgido numeras plataformas de conocimiento donde se comparten herramientas que permiten la resolución de los problemas que surgen durante el desarrollo.

Una de estas herramientas es la clasificación y búsqueda de proyectos similares al nuestro. El uso de meta-datos o palabras asociadas a un objeto se ha extendido en la red como método de clasificación, estas palabras son conocidas como tags y ayudan a buscar, compilar y contextualizar la información que abunda en la red.

Mi proyecto es un módulo que trabaja sobre una plataforma de conocimiento, donde el usuario aloja sus proyectos[1]. Este módulo consiste en localizar e insertar referencias de proyectos similares al subido por el usuario, buscadas en la web, concretamente en otras plataformas de desarrollo. Además los resultados son presentados al usuario de forma intuitiva a través de una navegación por etiquetas. Las consultas son desarrolladas sobre webs de buenas prácticas de desarrollo como son Codeplex y SourceForge.

La interacción con el usuario es ya una necesidad en cualquier proyecto que deseemos implementar sobre la tan citada web 2.0. La aplicación interactúa con el usuario, permitiéndole decidir que referencias se mostrarán asociadas a su proyecto a través de la aprobación de estas. Además podrá decidir las páginas y los motores de búsqueda donde se realizarán las consultas.

También resultaba importante que el proceso de búsqueda estuviera automatizado, esto se desarrolló complementando las búsquedas manuales y las automáticas, para las etiquetas de los proyectos. De este modo, tenemos una plataforma dinámica que interactúa con el usuario, el cual es capaz de servirse tanto de los proyectos alojados en la web como los desarrollados externamente a esta.

Para controlar el proceso de desarrollo utilizamos una metodología enmarcada dentro del concepto ágil. Estas metodologías ágiles tienen entre sus conceptos clave la aceptación del cambio, así el proyecto aunque se define en un inicio, se va redefiniendo conforme avanza el desarrollo y surgen los cambios inherentes a cualquier proceso. También algunos aspectos de este módulo se han definido conforme avanzaba el desarrollo, siempre siguiendo el criterio de la calidad continua, también propio del pensamiento ágil. El proceso concreto que usé se denomina Scrum. El proceso divide el trabajo en iteraciones denominadas Sprint.

Este proyecto lo desarrollé en el Centro Europeo de empresas e innovación de Navarra, también conocido como CEIN, más concretamente en el área de excelencia de Microsoft. Todo él se desarrolló sobre un servidor con SharePoint 2010, realizando los correspondientes programas con Visual Studio.



A lo largo de esta memoria desarrollaré los siguientes temas:

- La arquitectura del sistema empleado.
- Una descripción de las metodologías ágiles, concretamente de Scrum. Cómo se ha aplicado ésta al desarrollo.
- Las herramientas utilizadas para la creación del módulo, así como las pruebas de concepto realizadas para implementar las diferentes funcionalidades.
- Para cada Sprint o iteración describiré los requisitos seleccionados, las incidencias en el desarrollo. Haré un seguimiento de las diferentes tareas valiéndome de gráficas y mostraré el producto entregable propio de Scrum.

## 2.Arquitectura del sistema

En el siguiente dibujo muestro de forma muy esquemática como es la arquitectura del servidor que he montado en una máquina virtual. Responde al modelo básico.

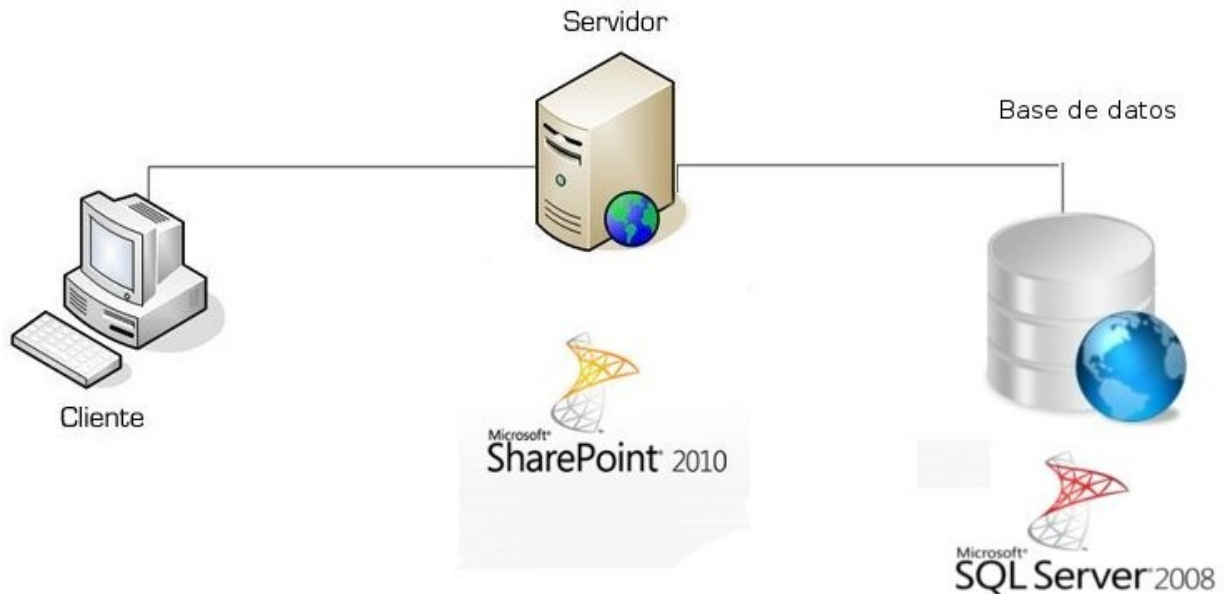


Figura 2.1 Arquitectura cliente servidor

En el lado del cliente podremos acceder a los sitios webs creados en SharePoint desde cualquier navegador, en nuestro caso trabajábamos en una red local de modo que cualquier otro ordenador de dicha red podía entrar a nuestra web con la URL.

Para montar el servidor lo hicimos desde una máquina virtual instalada en Windows 7 con un procesador de 64 bits, lo que nos permite aprovechar al máximo la memoria del ordenador. Así pudimos asignarle más memoria RAM a nuestra máquina virtual donde iba a estar alojado el servidor. La elección principal de la máquina virtual a utilizar era Virtual PC que pertenece a Microsoft, sin embargo no es posible alojar en virtual PC un sistema operativo de 64 bits, lo cual es requisito de instalación de SharePoint 2010. Al final nos decantamos por usar VMware Server, que inicialmente era de pago y más tarde fue gratuita, nos permite un gran manejo de los recursos para una o varias máquinas corriendo de forma simultánea.

En la figura 2.2 de la siguiente página podemos ver la pantalla de configuración de la central de VMware server[2], desde donde podemos crear máquinas virtuales, configurar sus características, ejecutarlas, pararlas o eliminarlas. Desde la pestaña de virtual machine podremos ver todas las máquinas de las que disponemos .

El sistema operativo que va hacer de servidor de mi aplicación es Windows Server 2008 R2, con procesador de 64 bits. Más tarde desarrollaré las incidencias de su instalación y de SharePoint 2010 en el sistema operativo, en el sprint de plataforma donde se montó el servidor. Cuando realicé el proyecto, SharePoint 2010 se encontraba todavía en fase beta.

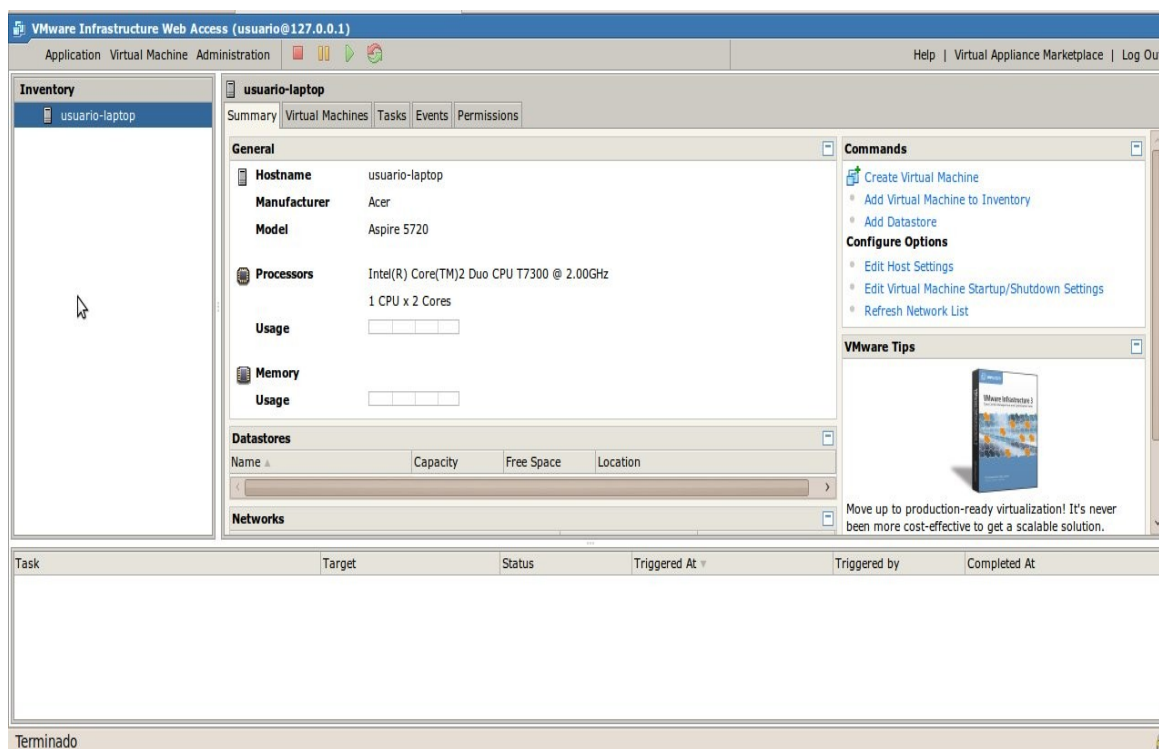


Figura 2.2 Pantalla de gestión de VMware server

Antes de definir la arquitectura interna de SharePoint definiré brevemente su significado. SharePoint es un conjunto de herramientas de administración de contenidos y procesos, al conjunto de herramientas se les denomina Productos y Tecnologías de SharePoint y ofrecen dicha funcionalidad a nivel empresarial. Entre sus productos disponen de Windows SharePoint Services conocido como WSS, y Microsoft Office SharePoint Server (MOSS), cuya última versión es la 2010. La diferencia entre ambos es que MOSS permite un mayor número de funcionalidades y también mejoras en las búsquedas, por lo cual esta es de pago y la otra no. Esta plataforma permite al usuario crear y administrar sitios para toda la organización, creando por ejemplo blogs o wikis, modificando el control a través de elementos webs, o gestionando librerías y listas. La estructura interna de SharePoint la podemos ver en la figura 2.3, este se apoya en SQL server. En el caso de SharePoint 2010, es compatible con SQL server 2008.

En dicha figura podemos ver la estructura interna de SharePoint, como para cada aplicación web, a través de IIS o los servicios de Internet (Internet Information Services), usa su propia base de datos de contenidos de SQL Server.

Dichas bases de datos de contenido almacenan todo el contenido del sitio Web, incluyendo documentos del sitio o archivos en bibliotecas de documentos, datos de listas, propiedades de WebParts, así como nombres de usuario y permisos.

Pero la base de datos más importante, y la que hace que interactúen las peticiones con las bases de datos adecuadas es la base de datos de configuración. Cuando el servidor de SharePoint recibe una petición de una página chequea en esta base de datos para determinar qué base de datos de contenido es la que almacena el fichero que se pide.

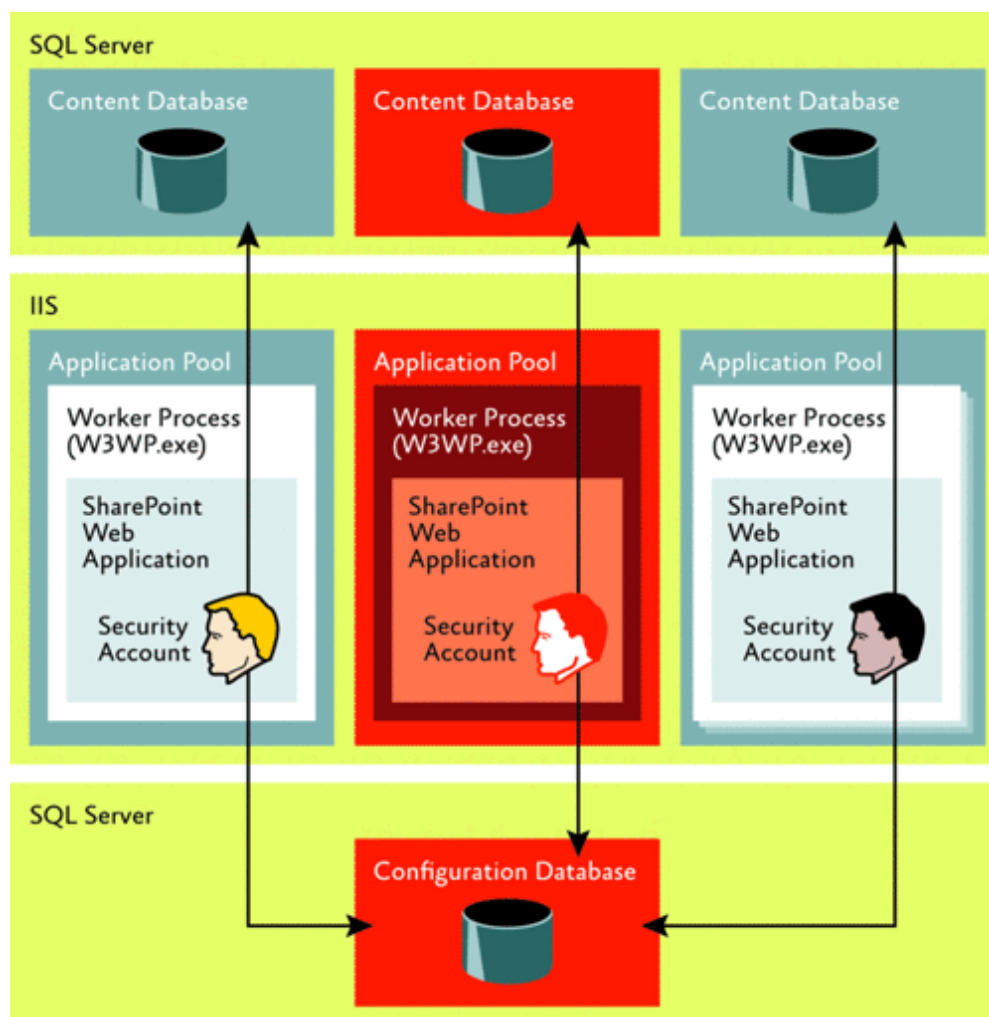


Figura 2.3 Estructura interna del manejo de bases de datos de SharePoint

A nivel de jerarquía SharePoint dispone de una serie de capas. La figura 2.4 muestra muy bien esta jerarquización. En el nivel superior está como es lógico el nivel hardware, los servidores físicos que componen la granja de servidores. El nivel de Aplicaciones Web es el primer contenedor en la jerarquía de SharePoint[3] (llamados en versiones anteriores Servidores Virtuales). Cada Aplicación Web puede ser vista como una instancia virtual del Portal, manteniendo su propio sistema de autenticación, puerto de IIS y configuraciones generales. Solamente a nivel de archivos físicos comparten las Aplicaciones Web algo en común. Cada Aplicación Web es un contenedor de Colecciones de Sitios. Las Aplicaciones Web solo pueden ser creadas desde la Administración Central, por programación o por medio de scripts.

El siguiente nivel es SPSite, o el sitio propiamente dicho, este puede ser creado desde la administración central, cada colección de sitios contiene al menos un sitio superior y luego diferentes subsitios asociados a él y, al igual que una aplicación web no posee una interfaz propia, será un contenedor de sitios webs con su propia autorización y configuración.

Un sitio web se puede ver como un subsitio de la colección de sitios creada, estos se pueden crear de forma dinámica y principalmente son contenedores de listas. Todos los tipos de listas derivan de las *listas personalizadas*, incluidas las librerías de documentos.

Además de todo esto si unimos las funcionalidades que nos proporciona Visual Studio 2010, podremos realizar controles de muchos tipos como las Visual WebParts, además de flujos de trabajo, paginas ASPX, elementos de Silverlight.. Este tipo de elementos se describirán en el capítulo herramientas y pruebas de concepto.

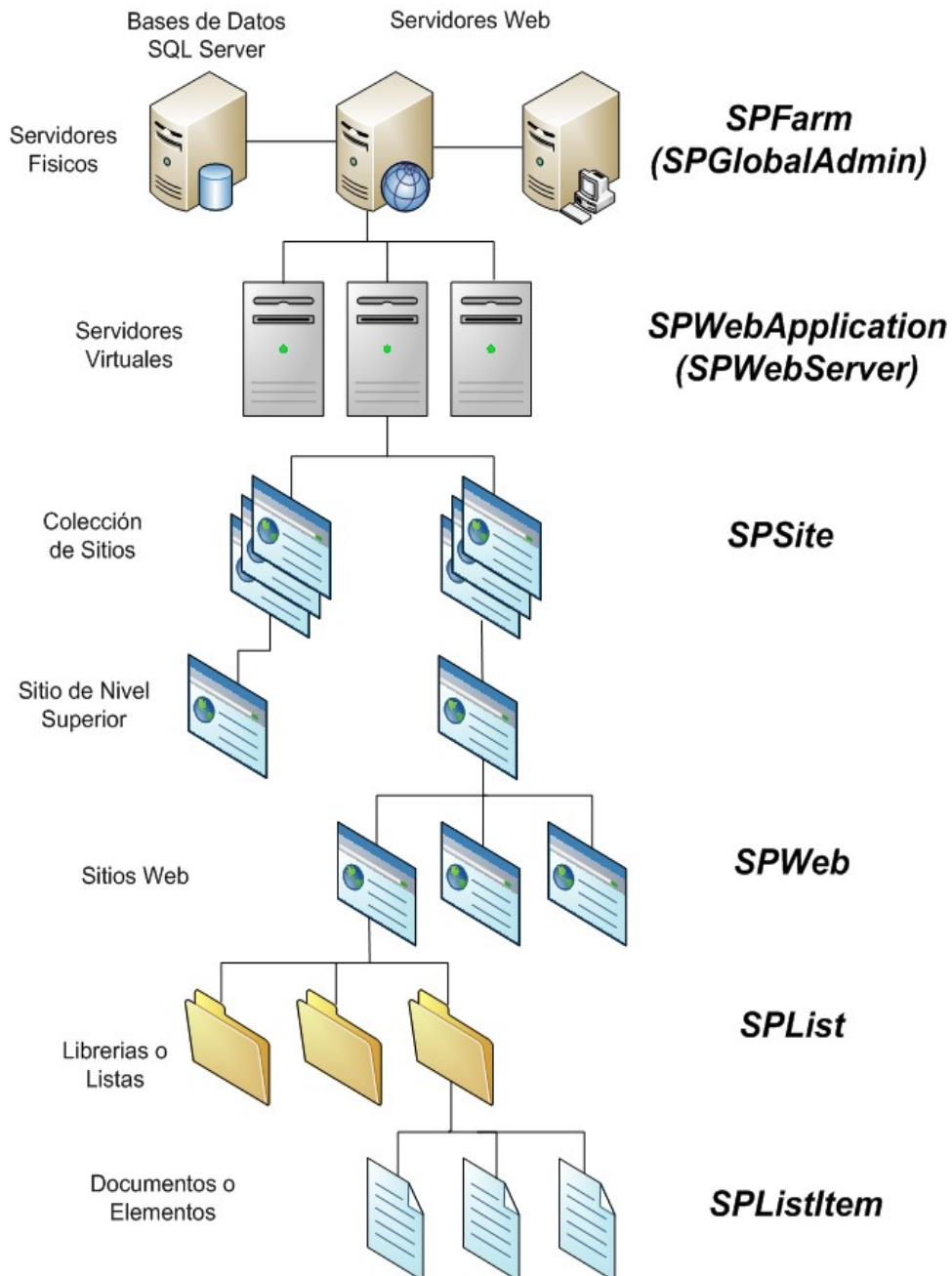


Figura 2.4 Jerarquía de SharePoint

### 3.Herramientas y pruebas de concepto

#### 3.1 Herramientas principales

La aplicación que he desarrollado se fundamenta en el framework de .NET, concretamente usamos el último, el framework 4.0, ya que se instalaba automáticamente con el programa Visual Studio 2010. En este último realizamos todos los controles correspondientes a nuestro sitio web, Visual Studio incluye varios lenguajes de programación, el que he usado principalmente ha sido C#, pero también XML y javascript.

Con las últimas versiones de SharePoint y Visual Studio, se ha logrado una mayor integración para el desarrollador, que permiten a este crear soluciones, características o flujos de trabajo para el sitio web de SharePoint de una manera muy sencilla e intuitiva. Lo que en versiones anteriores debíamos andar modificando varios archivos y realizar instalaciones a través de comandos, ahora podemos compilar, exportar o importar una solución a SharePoint con sus características, listas, archivos o lo que fuere, pulsando una sola tecla.

También he usado de manera menos habitual SharePoint Designer 2010, este programa es un editor de páginas web para SharePoint, nos permite crear y modificar páginas. Podemos además hacer flujos de trabajo, que explicaré a continuación, todo ello usando una interfaz gráfica e intuitiva.

#### 3.2 WorkFlow

WorkFlow o flujo de trabajo es la automatización de una secuencia de tareas o actividades, que se desarrolla cuando ejecutamos un proceso, guardando el estado de cada etapa del proceso.

Los flujos de trabajo en SharePoint están asociados a la creación o modificación de uno o varios elementos de una lista o librería. También se pueden crear flujos de trabajo rápidamente sólo para ocuparse de procesos habituales del sitio de su equipo, por ejemplo, enviar notificaciones o crear tareas.

Existen varias formas de configurar un flujo de trabajo en SharePoint, en primer lugar para flujos sencillos de seguimiento o aprobación, podremos desde la propia configuración de nuestro sitio seleccionar flujos predefinidos para un elemento o un documento de una lista específica. Para definir estos flujos de forma gráfica e intuitiva está SharePoint Designer. Sin embargo la configuración está limitada, si deseamos crear flujos complejos y con código personalizado deberemos usar Visual Studio.

En mi proyecto, era necesario crear un flujo que buscara referencias para los proyectos subidos. A la hora de definir las etiquetas de un proyecto, doy la posibilidad al usuario de buscar de forma manual referencias en las webs que considere. El usuario también puede añadir etiquetas sin buscar manualmente, es entonces cuando el flujo se lanza y busca referencias para las etiquetas que no tengan ninguna en el registro de búsquedas. Realizar esto sólo era posible mediante programación, el diagrama del flujo implementado se muestra en la figura 3.1.

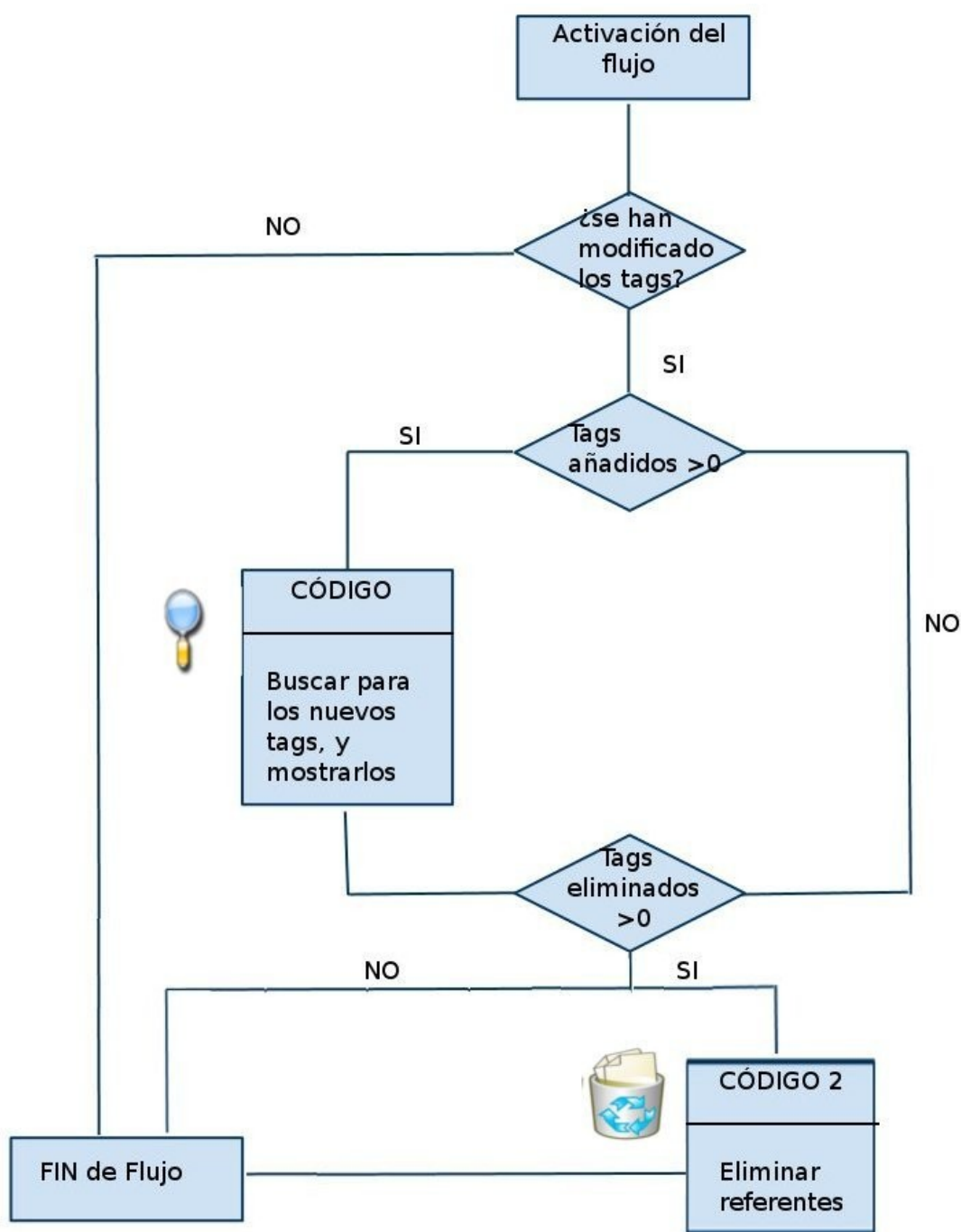


Figura 3.1 Diagrama de Flujo de trabajo

El Flujo se activa cuando se modifica un proyecto. Cuando éste ocurre, lo primero que hago es una comparación. Si se han modificado las etiquetas (también denominados tags a lo largo de la memoria), paso a comprobar que tags se han añadido y eliminado, en caso contrario el flujo finaliza y no devuelve nada.



Todo este flujo trabaja sobre la lista de proyectos donde he definido las siguientes columnas: *Título*, *ID del proyecto*, enlace a la librería (ya que en la librería se guardan todas las carpetas y documentos del proyecto), *Tags* (las etiquetas) e *Info de Tags*. En esta última columna es donde devuelvo la información recogida por el flujo al usuario.

El primer cuadrado del diagrama de código es donde para cada tag añadido hago lo siguiente. En primer lugar, compruebo si ya se han buscado alguna vez referencias externas para dicho tag. Para ello sólo tengo que mirar en una lista donde guardo todas las referencias buscadas. En el caso de que ya existan, devuelvo en la columna de información una URL a mi página de resultados para esa referencia. Más adelante explicaré como funciona la navegación de resultados a través de la nube de etiquetas.

Ahora bien, si no existe, lo que significa que no se han buscado previamente referencias para esa etiqueta, entonces busco para ese tag referencias o proyectos similares en las webs de buenas prácticas que guardo en la lista de *webs favoritas* elegidas por el usuario. Si el usuario no ha navegado aún y no ha elegido sus webs favoritas en las búsquedas manuales, el flujo automáticamente hará una consulta con HTML a la web de Codeplex.

El primer bloque finaliza cuando se han analizado todos los tags o etiquetas añadidos al proyecto. En el caso de que se hayan eliminado uno o más tags, para cada uno de ellos se ejecutará el siguiente bloque del flujo de trabajo (CODIGO 2). Eliminaré todos los referentes de una etiqueta, cuando esa etiqueta no la comparta ningún otro proyecto alojado en la plataforma. De esta manera cuando borro una etiqueta única se eliminarán automáticamente todas sus referencias del registro de búsquedas. Así si se vuelve añadir en otro proyecto, se buscarían de nuevo con el flujo. Esto provoca que no se almacenen datos inservibles, haciendo la plataforma más dinámica. Las incidencias de la creación del flujo las detallaré en el sprint de automatización de búsquedas. A continuación explicaré las pruebas de concepto y las herramientas que he usado para la realización de las búsquedas.

### 3.3 Búsquedas

Uno de los aspectos más importantes de mi proyecto era la capacidad de buscar en webs de buenas prácticas. Para ello realicé pruebas de concepto con el fin de conocer la mejor manera de realizar búsquedas desde SharePoint y obtener los resultados planeados con el cliente (en este caso mi tutor). El objetivo, en primer lugar, era extraer información de páginas concretas. Las indicadas eran en principio Codeplex y SourceForge. Tras ponerme en contacto con dichas webs se me afirmó que no tenían ningún servicio web de búsqueda interna público.

Los servicios web son una manera de comunicar aplicaciones web, uno de los más importantes es SOAP, que usa Xml y contiene descripciones en WSDL. Este funciona muy bien en aplicaciones interconectadas que transmiten poca información simultánea. SOAP que se construyó como la solución definitiva, ha sido últimamente criticada y puesta en entredicho tras la aparición de REST. Este último se basa principalmente en Http, tal y como la propia web, y su sencillez lejos de ser un inconveniente parece una ventaja.

En mi caso las pruebas de concepto realizadas fueron sobre web services SOAP, viendo que Google dejó de dar soporte a su api Google SOAP Api Search en 2006, las búsquedas finalmente las implementé de dos maneras diferentes, a las que definí como



consultas Html y consultas Xml. Para gestionar estas búsquedas en mi aplicación web, creé una clase llamada gestor de búsquedas que se encargaba de dirigir las.

### 3.3.1 Búsquedas con Html

Las primeras búsquedas que implementé las defino como consultas HTML, principalmente, porque se basan en leer de este lenguaje. Tan sencillo como acceder al código fuente de la página de búsquedas de proyectos de Codeplex o SourceForge, y leerlo desde .Net como si de strings se trataran. Para buscar la etiqueta o etiquetas correspondientes solo hay que indicarlo en una query de la URL. En la captura de pantalla de la figura 3.2 muestro la página de búsquedas de proyectos de Codeplex sobre la que realizó la búsqueda

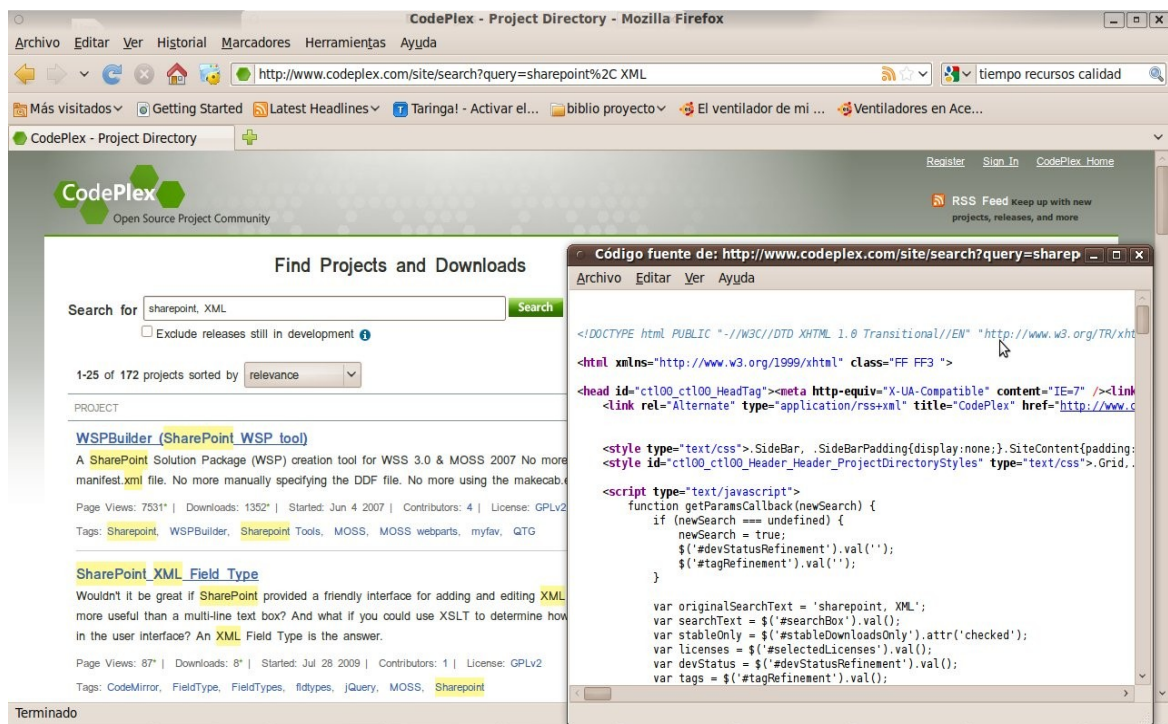


Figura 3.2 Captura de Codeplex y su código fuente

En el caso de ejemplo he buscado para *SharePoint, XML*, viendo la URL vemos que puedo buscar también indicándolo en ésta. La primera función a elaborar era la que devuelve el código fuente de cualquier web. Valdría tanto para Codeplex como SourceForge.

```
static private StreamReader leerPaginaWeb(string laUrl)
{
    // Cear la solicitud de la URL.
    HttpRequest request = (HttpRequest)WebRequest.Create(laUrl);

    // Obtener la respuesta.
    HttpResponse response = (HttpResponse)request.GetResponse();

    // Abrir el stream de la respuesta recibida.

    Console.WriteLine("Petición aceptada");
    Console.WriteLine("");
}
```

```

Console.WriteLine("");
StreamReader reader = new StreamReader(response.GetResponseStream());
"

// Leer el contenido.

// string res = reader.ReadToEnd();

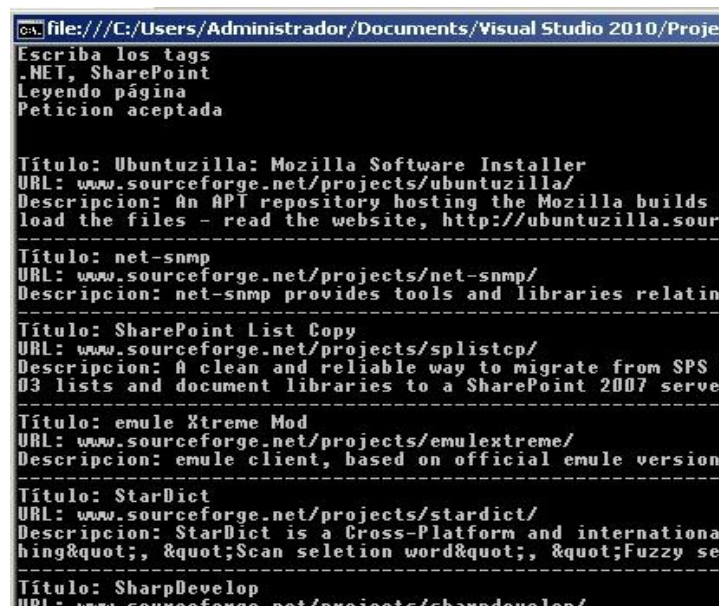
// Mostrarlo.

return reader;
// Cerrar los streams abiertos.

}

```

Tras obtener el HTML y analizarlo, a través de las funciones de manejo de strings, recupero los datos que me interesan guardar para posteriormente registrarlos en SharePoint. En la siguiente imagen se puede ver como las pruebas de concepto para SourceForge funcionaban correctamente devolviendo el contenido de la página web.



### 3.3 Prueba de búsqueda en SourceForge

Una vez obtenidas las referencias para las etiquetas escritas por el usuario las guardo en una tabla de resultados, que posteriormente se registrará en la lista de mi sitio de SharePoint llamada *Registro Búsqueda* que incluye los campos: Título, descripción, tags de la referencia y aprobación de esta entre otras.

Siempre que registro un resultado, la columna de aprobación se inicializa automáticamente como *Pendiente*. De este modo es el usuario quien debe aprobar las referencias encontradas para cada etiqueta.

El método registrar() es común tanto a las búsquedas en HTML que he definido para Codeplex y SourceForge, como para las búsquedas XML que he implementado y describo a continuación. Este método comprueba que la referencia es nueva llamando a la función *isRegistered()*. Uso una tabla de *filas*, una estructura que he creado que equivale a una referencia.

```
public int registrar(fila[] búsqueda, string tags)
{
    // Aquí me conecto a mi sitio de SharePoint

    SPSite currentSite = SPControl.GetContextSite(HttpContext.Current);
    SPWeb miWeb = currentSite.OpenWeb();
    int x = 0;
    int cont = 0;
    SPListItemCollection items = miWeb.Lists["Registrobúsqueda"].GetItems();

    // Recorro los resultados de la búsqueda y guardo los únicos en el registro

    while (resultado[x].titulo != "FIN")
    {
        if (!isRegistered(resultado[x].titulo, items))
        {
            SPListItem item = items.Add();

            item["Título"] = búsqueda [x].titulo;
            item["URL"] = búsqueda [x].URL;
            item["descripcion"] = búsqueda [x].descripcion;
            item["posibles tags"] = tags;

            item.Update();
            cont++;
        }

        x++;
    }

    return cont;
}
```

### 3.3.2 Búsquedas con Xml

Tras barajar varias posibilidades y pensando inicialmente en realizar una búsqueda usando los web services SOAP, acabé descubriendo un método más sencillo y efectivo de realizar búsquedas sobre cualquier sitio web leyendo simplemente de un Xml. Los web services que propone Bing resultaron funcionar correctamente en las pruebas de concepto en Visual Studio, y con ellos podía realizar cualquier búsqueda de manera ágil, sin embargo a la hora de subir este código a un WebPart en SharePoint, los errores de permisos comenzaron a surgir.

Las búsquedas que implanté al final fueron las dos sobre XML, usando los motores de Bing y de Yahoo. Con Google no me fue posible realizar búsquedas personalizadas, ya que las soluciones gratuitas que el buscador ofrece incluyen toda publicidad. Los pasos a seguir para ambos buscadores son similares, así que describiré tan sólo uno: las búsquedas de XML en Bing[4].

Lo primero que necesitamos es una cuenta en Windows Live, con ella crearemos una AppId en la página de desarrolladores de Bing : <http://www.bing.com/developers/>, a esta Id le asociamos la página en la que vamos a usar los servicios que dispone el buscador (SOAP, XML, JSON...). En la figura 3.4 podemos ver como quedaría nuestra aplicación creada desde la página de Bing.

**Bing API 2.0**

A key part of Project Silk Road is a re-architected Bing API that offers open, flexible options for building or enhancing your site or applications. Developing an application with the new API is straightforward: Choose a SourceType (or SourceTypes—you're not limited to one), choose an output protocol (JSON, SOAP, or XML) and then customize according to your needs.

[Get started by applying for an AppID now.](#)

[Disable](#) · [Enable](#) · [Delete](#) [Select all](#) · [Clear all](#)

Application name and AppID	Website	
<input type="checkbox"/> miApplication F216C49337B833B66A684083C08DF1B9A6CAE891	http://win-ivgatwi52b:20294	<a href="#">Copy</a>

© 2010 Microsoft | [Privacidad](#) | [Legal](#) | [Anuncios](#) | [Ayuda](#) | [Comentarios](#)

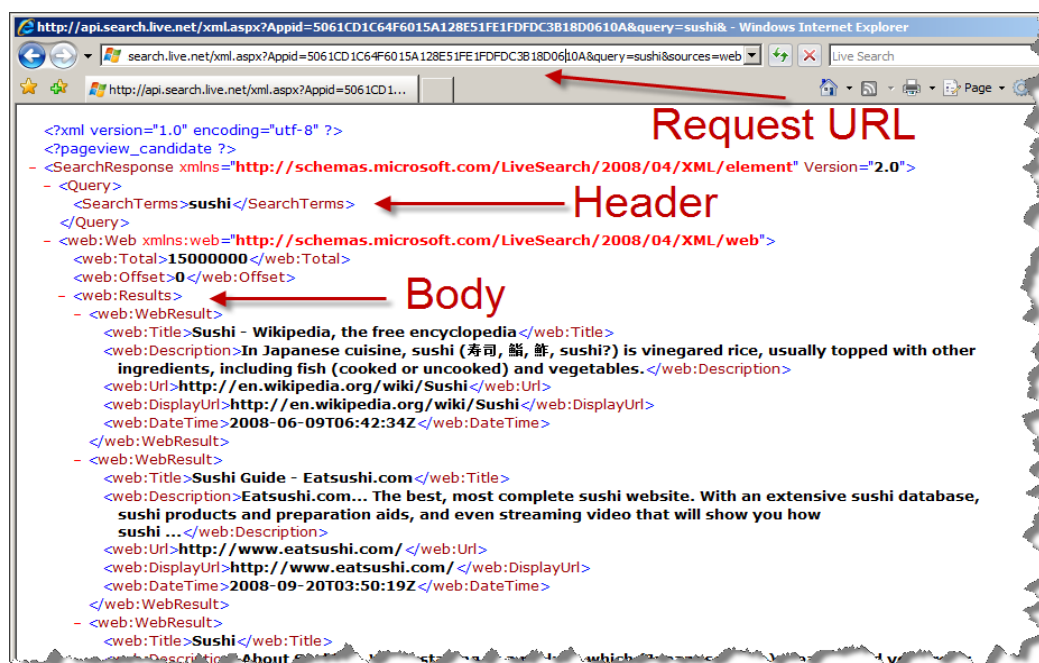
Figura 3.4 AppId de Bing

Ahora ya podemos acceder a las búsquedas en XML usando la siguiente URL:

**`http://api.search.live.net/xml.aspx?Appid=<Nuestro AppId>&query=<Lo que queremos consultar>`**

En el parámetro query si queremos buscar para más de una palabra hay que separar cada uno por '&', se pueden configurar las búsquedas de muchas maneras. Por ejemplo con *Sort by* ordenamos según los parámetros que consideremos, o la que uso yo para diferenciar webs con *&site=* elegimos la página en la que buscar la query valiéndose del motor Bing.

El resultado sería un página con los resultados mostrada en XML tal y como se muestra en la figura 3.6, sólo queda leerla y analizar los datos para poder registrar los datos en nuestra lista de SharePoint. La función de registrar es la misma que la usada en consultas HTML ya que cuando devuelvo los datos de la búsqueda lo hago en una tabla de mismas características.



**Request URL**

**Header**

**Body**

```
<?xml version="1.0" encoding="utf-8" ?>
<?pageview_candidate ?>
- <SearchResponse xmlns="http://schemas.microsoft.com/LiveSearch/2008/04/XML/element" Version="2.0">
- <Query>
  <SearchTerms>sushi</SearchTerms>
</Query>
- <web:Web xmlns:web="http://schemas.microsoft.com/LiveSearch/2008/04/XML/web">
  <web:Total>15000000</web:Total>
  <web:Offset>0</web:Offset>
- <web:Results>
- <web:WebResult>
  <web:Title>Sushi - Wikipedia, the free encyclopedia</web:Title>
  <web:Description>In Japanese cuisine, sushi (寿司, 鮓, sushi?) is vinegared rice, usually topped with other ingredients, including fish (cooked or uncooked) and vegetables.</web:Description>
  <web:Url>http://en.wikipedia.org/wiki/Sushi</web:Url>
  <web:DisplayUrl>http://en.wikipedia.org/wiki/Sushi</web:DisplayUrl>
  <web:DateTime>2008-06-09T06:42:34Z</web:DateTime>
</web:WebResult>
- <web:WebResult>
  <web:Title>Sushi Guide - Eatsushi.com</web:Title>
  <web:Description>Eatsushi.com... The best, most complete sushi website. With an extensive sushi database, sushi products and preparation aids, and even streaming video that will show you how sushi ...</web:Description>
  <web:Url>http://www.eatsushi.com/</web:Url>
  <web:DisplayUrl>http://www.eatsushi.com/</web:DisplayUrl>
  <web:DateTime>2008-09-20T03:50:19Z</web:DateTime>
</web:WebResult>
- <web:WebResult>
  <web:Title>Sushi</web:Title>
  <web:Description>About...
```

Figura 3.5 Resultado de consulta XML con Bing

De esta manera sólo queda integrar ambas búsquedas en mi sitio de SharePoint. Para ello en la WebPart de subida de proyectos, añadí una clase gestora de búsquedas desde la cual con los controles establecidos manda buscar según las peticiones del usuario. Todo esto es mucho más visible viendo la interfaz, es el caso de la captura 3.6 que corresponde a una vez subido un proyecto a la plataforma.

En esta captura podemos apreciar como el usuario posee una serie de opciones una vez subido el proyecto. Si lo desea puede añadir etiquetas o tags, lo cual es recomendable, y para estas etiquetas puede buscar referencias de forma manual. Como ya he comentado antes si sólo eligiese añadir el flujo buscaría referencias para los nuevos tags de forma automática, y eliminaría los borrados. Sin embargo, si elige marcar la casilla de buscar referencias se activarán las opciones de búsqueda.

Figura 3.6 Captura de proyecto subido

Una vez escritas las etiquetas y marcada la casilla de búsqueda, podremos realizar búsquedas simultáneas para estas etiquetas. Podemos elegir entre los motores de búsqueda tres diferentes; uno correspondiente a las búsquedas HTML llamado *HTML Custom*, que nos dará la opción de buscar en SourceForge y en Codeplex, los otros dos correspondientes a las búsquedas XML. En estos últimos podemos elegir en qué páginas queremos buscar usando el motor de Yahoo o Bing o ambos. En la captura de búsqueda realizada podemos ver que el último motor seleccionado ha sido Bing que incluye la posibilidad de elegir página. Cuando marcas esta opción permito al usuario a partir de un nuevo textbox y un botón añadir la página sobre la cual el motor Bing va a buscar. Antes de almacenar esta página compruebo que se pueda acceder a ella.



El proyecto se ha subido correctamente. Para visitarlo pulse [aquí](#).

[Volver a subir](#)

Añada los tags separados por comas

Marque la casilla si desea buscar referencias externas ☒

Elija un motor de búsqueda

Pulse el boton '>' para añadir la pagina a la búsqueda

elegir pagina..	X	SourceForge (HTML)
<a href="http://www.codeplex.com">www.codeplex.com</a>	>	Yahoo: <a href="http://www.codeplex.com">www.codeplex.com</a>
<a href="http://www.codeproject.com">www.codeproject.com</a>	<	Bing: <a href="http://www.codeproject.com">www.codeproject.com</a>

Buscando en HTML custom los tags: Sharepoint, XML, con SourceForge (HTML)  
25 referencias encontradas  
23 referencias registradas  
Buscando en [www.codeplex.com](http://www.codeplex.com) los tags: Sharepoint, XML, con Yahoo  
10 referencias encontradas  
6 referencias registradas  
Buscando en [www.codeproject.com](http://www.codeproject.com) los tags: Sharepoint, XML, con Bing  
10 referencias encontradas  
9 referencias registradas

Tags añadidos correctamente

[38 referencias registradas](#)

Figura 3.7 captura de búsqueda realizada

La línea final es además un vínculo al registro de búsquedas, ordenado por las últimas referencias creadas, desde el cual podemos editar las referencias.

## 4. Metodología de desarrollo aplicada: Scrum

### 4.1 Metodologías ágiles

El elevado índice de proyectos software que no llegan a buen puerto, bien por falta de tiempo o presupuesto, unido al constante crecimiento de internet y las tecnologías hacen necesario la implantación de un modelo de desarrollo capaz de adaptarse a los inevitables cambios.

El concepto de ágil no es algo nuevo, ni propio del software. Desde la época de la producción en masa, encabezada por Henry Ford, han surgido numerosos métodos de producción. Uno de ellos, precursor de lo que ahora conocemos como *ágil* fue la corporación creada por Kiichiro Toyoda, conocido como Toyota[5].

Dicha corporación se enfrentaba a un mercado saturado y heterogéneo, en una economía de posguerra con un espacio reducido y caro. La clave de su éxito se basa en conceptos como la mejora continua del proceso, la maximización del valor devuelto al cliente ( en frente a la cantidad, que imperaba en la producción en masa) y por tanto eliminación del desperdicio, bien sea el movimiento innecesario de personas, las esperas, incluso el inventario; que no aportan ningún valor al cliente.

Todos estos valores son aplicables también al desarrollo de software , así varios años más tarde en marzo de 2001, un grupo de expertos en modelos de mejora de desarrollo de software publicaron un texto conocido como el manifiesto ágil, que decía lo siguiente:

*Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar:*

- *A los **individuos y su interacción**, por encima de los **procesos y las herramientas**.*
- *El **software que funciona**, por encima de la **documentación exhaustiva**.*
- *La **colaboración con el cliente**, por encima de la **negociación contractual**.*
- *La **respuesta al cambio**, por encima del **seguimiento de un plan**.*

*Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.*

La última frase es importante, no se debe asociar una metodología ágil por ejemplo a la falta de documentación, el paradigma dice que esta es importante sin embargo se le da mucha más importancia a que el software funcione, porque es lo que aporta valor real al cliente.

Existen varios métodos de desarrollo ágil, y todos ellos cumplen estos conceptos. Por eso es importante asimilarlos si queremos implantar dichos modelos. Si no colaboramos con el cliente o no tenemos capacidad de responder a un cambio no estaremos siendo ágiles no importa lo que hagamos. Ejemplo de metodología ágil es la que he implantado en mi proyecto, Scrum[6], de la que hablaré a continuación.

## 4.2 Scrum

Como ya he explicado previamente Scrum es una metodología de desarrollo que se enmarca dentro de las ágiles, es el resultado de la mejora continua del proceso. El significado de la palabra proviene del rugby, scrum significa melé; esa figura en la que los compañeros del equipo se amontonan, forman una pila y empujan todos en la misma dirección. Es fundamental la transparencia en la comunicación del equipo, que se organicen y motiven entre ellos. Todo ello siguiendo la premisa de maximización del valor, valor marcado por el cliente por lo que es fundamental que el cliente se integre en el proceso de desarrollo. Scrum se caracteriza por entregas frecuentes de productos funcionales al cliente, haciéndole a éste mucho más participe en el proceso.

Esta participación del cliente, la unión del equipo y las entregas a corto plazo, hacen que el proyecto en desarrollo sea más flexible a los cambios. Los requisitos del proyecto no serán una pieza inamovible como se entendían en los modos de desarrollo clásico. Esto sigue la premisa de que el cambio es inherente a cualquier desarrollo de una aplicación compleja, se trata de tomar el cambio como algo positivo que otorga valor al producto. Todo esto sin olvidar nunca la calidad, ésta se consigue cada día, la calidad debe ser la meta del equipo.

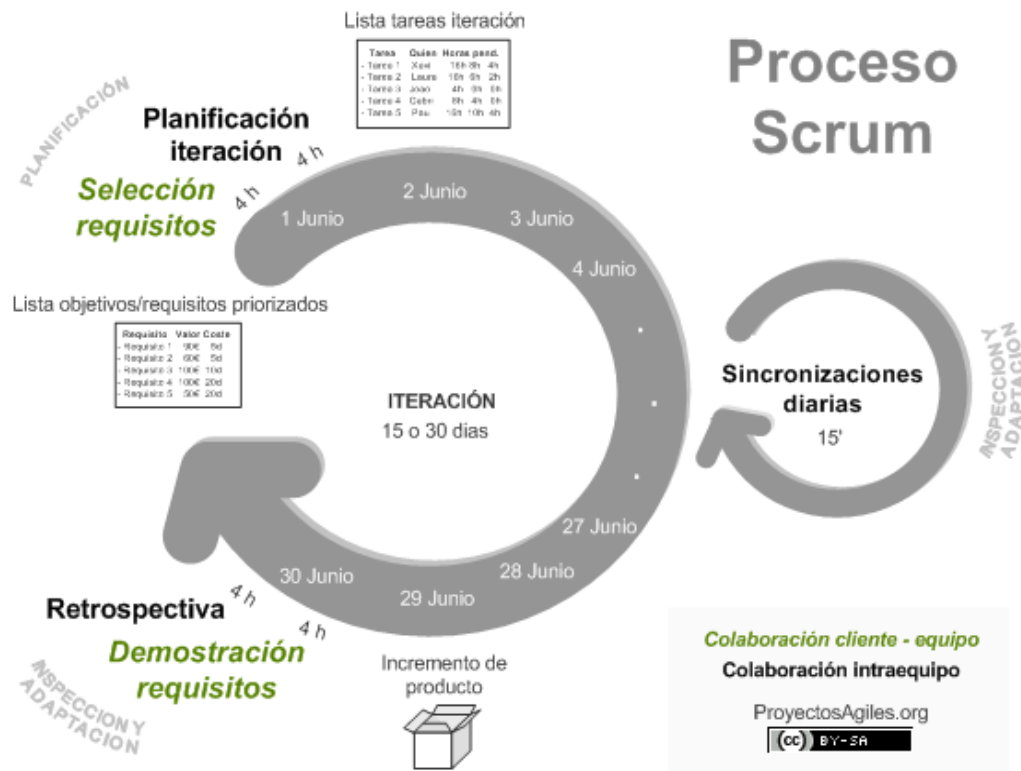
La recogida de requisitos para crear un producto se realiza teniendo en cuenta la visión del cliente y del usuario. Para ello se utilizan las historias de usuario, unas sencillas tarjetas en las que se recoge de forma esquemática y en un lenguaje claro qué es lo que queremos hacer. Con esas historias de usuario construimos la lista de requisitos del producto o *product backlog*. A cada ítem de la lista se le asigna una prioridad. El equipo tiene que estimar cuánto tiempo es necesario para realizar cada una de las tareas.

El proceso se desarrolla mediante iteraciones, estas son cortas y fijas, suelen ser desde dos semanas hasta un mes natural, cada iteración produce un incremento del producto a través de entregables. Se denominan Sprints.

Los roles que se asumen se pueden dividir en dos, de manera informal y sin ánimo de ser despectivo se suele llamar “cerdos”, a los que están verdaderamente comprometidos con el proyecto, estos son el *product owner* (cliente), el *scrum master* (el papel de facilitador se impone al de jefe) y el equipo (unido y formado en diferentes disciplinas). Luego están los roles más secundarios que también hay que tener en cuenta, conocidos como los “pollos” estos son usuarios, clientes y vendedores, directivos.

Podemos observar como se ejecutan las iteraciones de Scrum en la figura 4.1 de la siguiente página.





#### 4.1 Diagrama de proceso de Scrum

Las actividades llevadas a cabo por iteración son las siguientes:

- **Planificación de la iteración:** En esta fase el cliente y el equipo se reúnen, el cliente define la lista de requisitos priorizada en función del valor. Luego el equipo planifica la iteración, estimando el esfuerzo necesario para cumplir con los requisitos comprometidos. Los miembros del equipo se auto asignan las distintas tareas que surgen.
- **Ejecución de la iteración:** El equipo se reúne diariamente unos 15 minutos con el fin de sincronizar las tareas a realizar dicho día. El facilitador o scrum master se encarga de eliminar los obstáculos y reducir las interferencias externas.
- **Inspección y Adaptación:** Aquí se realizan la demostración y la retrospectiva. La primera se trata de presentar al cliente los requisitos cumplidos en forma de incremento de producto, en función de esto el cliente podrá replantear sus prioridades y objetivos de cara a las nuevas iteraciones. En la retrospectiva es el equipo quien se reúne para resolver los problemas que han surgido durante la ejecución e imposibilitan la continuidad, donde el facilitador tratará de ayudar todo lo posible.

El uso de material gráfico ayuda mucho a la motivación y realización del proyecto, el equipo cuenta con una pizarra donde podrán seguir la evolución de las distintas tareas. Además existen otras herramientas como gráficas o las cartas de estimación.

En definitiva, Scrum como todo proceso de mejora se basa en proporcionar una mayor productividad al desarrollo software, y lo hace corrigiendo errores del pasado, centrándose en el valor del producto entregado. Valiéndose de conceptos tan importantes

como la colaboración del cliente, la comunicación del equipo, la calidad continua, la motivación... Personalmente, tras mi experiencia desarrollando este proyecto, creo que Scrum aporta muchas ventajas y facilidades. La implementación de este proceso ha sido, en mi caso, un tanto particular puesto que el equipo era prácticamente yo sólo, pero la colaboración con mi entorno de trabajo hizo que no lo notase durante el desarrollo, puesto que siempre tenía con quien consultar cualquier duda en todo momento. A continuación desarrollaré como hemos aplicado Scrum.

#### 4.3 Aplicación de Scrum al proyecto

Este módulo de inserción y localización de referencias lo he desarrollado en gran parte gracias a que gente muy experta me ha ayudado en todo momento. El balance que hago de usar Scrum como proceso es muy positivo. Si algo es fundamental en este proceso es la comunicación, tanto con el equipo, como con el cliente o con el facilitador; y en mi caso la ha habido durante todo el desarrollo.

Nuestros sprints duraban 3 semanas, como ya he explicado antes el sprint es el intervalo que dura una iteración donde se crea un incremento del producto. La singularidad era que en este caso el equipo era de una sola persona. Sin embargo tanto mi compañero que ha implementado otra funcionalidad para la plataforma complementaria a la mía, como el resto de compañeros que supervisaban el correcto desarrollo, hacían las veces de equipo ante los problemas que surgían. Los roles designados resultaban de la siguiente manera: el producto owner sería mi coordinador del proyecto, Jesús Villadangos, que se involucró semanalmente en el proceso, el scrum master o facilitador sería Rafael Flores el jefe de los Centros de Excelencia de Microsoft, aunque este papel podría ser repartido con mis otros dos compañeros que realizaron un seguimiento de los dos módulos que realizamos a lo largo del año: Raul Mayo y Goretti Ortigosa.

Al comienzo del proceso definimos con el cliente las distintas historias de usuarios que debía implementar el producto. Tras ello realizamos con el Scrum Master el producto backlog con las estimaciones en esfuerzo para cada historia de usuario. El resultado del product backlog se puede observar en la tabla 4.2. Éste es una declaración de objetivos, no es algo inamovible, de hecho a lo largo de las diferentes reuniones semanales y retrospectivas fue variando, añadiendo nuevas historias de uso y priorizando otras.

Índice	Historia de Usuario	Importancia	Estimación
A	Plataforma / Set up	40	18h
B	Gestión de servicios	70	30h
C	Consultas HTML	100	48h
D	Consultas XML	100	54h
E	Estructura de resultados	80	24h
F	Generalización y análisis	80	30h
G	Automatización búsquedas	90	10h
H	Integración	60	30h

#### 4.2 Product Backlog

Con el product backlog ya priorizado sólo queda para cada reunión antes de sprint elegir que historias entrarán en cada uno. Más tarde dividíamos cada historia en pequeñas tareas, usando una pizarra para que sea más gráfico. La pizarra se divide en tareas sin empezar o *non started*, en progreso o *in progress*, acabada o *done* y checked, cuando la tarea ha sido revisada. Podemos ver un ejemplo de como queda la pizarra un día de trabajo en el sprint 4, en la figura 4.3.



Figura 4.3 pizarra de trabajo de Scrum

Se puede observar en la fotografía que a la izquierda tenemos las historias de usuario estimadas, Scrum nos propone que estimemos en puntos de esfuerzo, una medida basada en el conocimiento de como trabaja el equipo, por ser equipos individuales y pioneros en este proceso decidimos estimar en horas. De las ocho historias de usuario iniciales en la fotografía, correspondiente al sprint 4, hay diez historias, como ya he explicado el product backlog es algo que puede variar a lo largo del desarrollo.



## 5. Desarrollo: Sprint 1

### 5.1 Selección de requisitos

Una vez realizado el product backlog, en función de éste decidimos al inicio de cada Sprint qué historias de usuario vamos a incluir. Para el primero era imprescindible desarrollar la plataforma, ya que es la base sobre la que vamos a trabajar. También debíamos definir los servicios a realizar, por tanto las historias de usuario Plataforma y Gestión de servicios se incluyeron por dependencias con otras historias. Aún quedaba tiempo para realizar otra historia de usuario o parte de ella. Así incluimos las Consultas HTML debido a la importancia que tenía asignada. De este modo preparamos nuestra herramienta principal de seguimiento con una primera división inicial de las tareas a realizar, reflejado en las figuras 5.1 y 5.2.

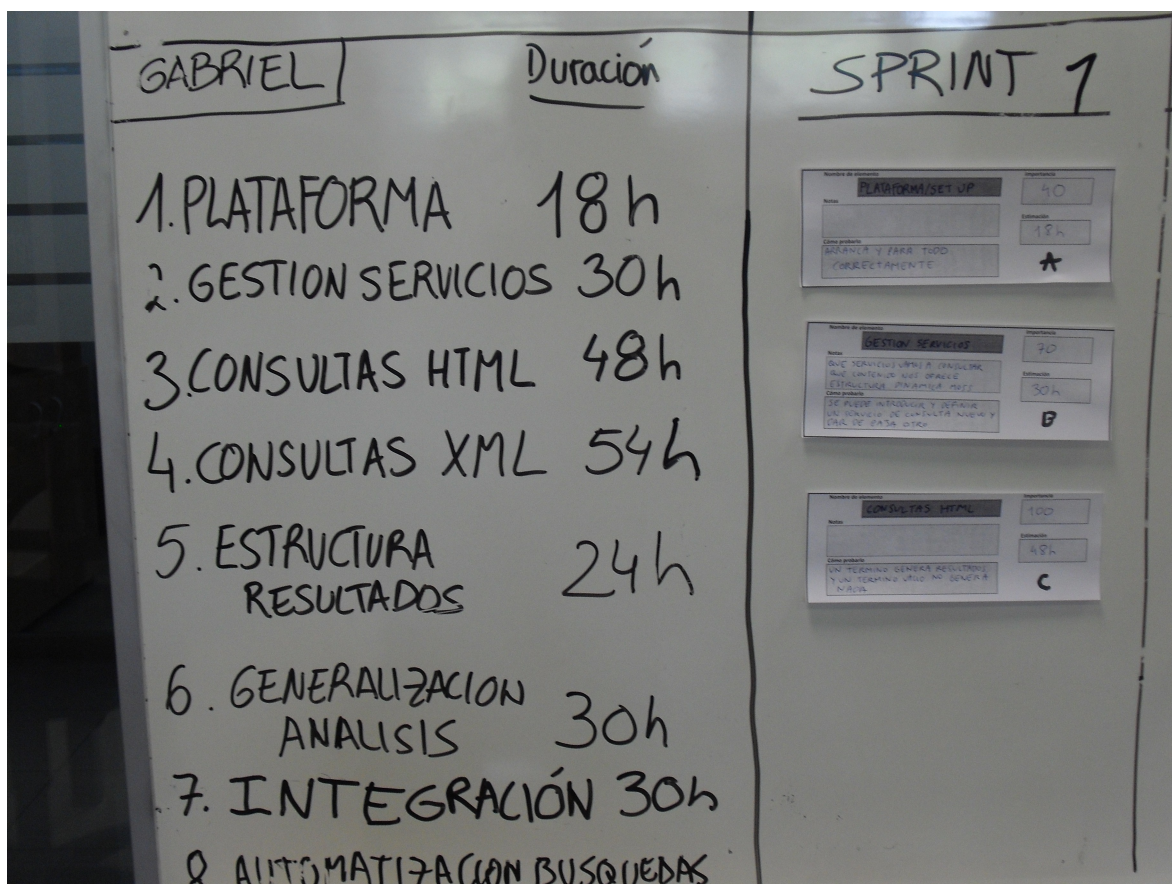


Figura 5.1 pizarra de inicio de Sprint 1 parte izda

La Plataforma engloba la instalación y configuración de la maquina virtual, que como podemos ver en la parte derecha de la foto inicial de la pizarra (fig. 5.2) se encontraba en *done*, ya que la habíamos instalado previamente. Otro aspecto de la plataforma es la instalación de SharePoint 2010 y las herramientas explicadas en el capítulo tres (SharePoint Designer, Visual Studio 2010). También en esta historia creé el sitio donde iba a desarrollar la aplicación web.

En el momento de realizar el product backlog, se definió la historia de usuario Gestión de servicios como la definición de los servicios que se iban a dar (pensando en realizar las búsquedas en una serie de webs predefinidas y no en todas como al final resultó), así como los resultados que se iban a devolver. También formaba parte de esta historia de usuario, la organización o estructura dinámica, de los servicios realizados en MOSS. Esto incluye la navegación a través de la nube de etiquetas y la creación de almacenes y listas donde se iban a guardar los datos.

Las consultas HTML las incluimos en el Sprint. Según la estimación inicial no se acabarían hasta el siguiente, ya que la suma de las tres historias hacen 96 horas, y disponemos de unas noventa horas por Sprint. Sin embargo este dato ha sido muy variable ya que en horas no todos los Sprint han durado lo mismo en función de las fechas. El resultado final fue favorable ya que en el tiempo disponible pude realizar las tres historias de usuario. No siempre ocurrirá así a lo largo del proyecto.

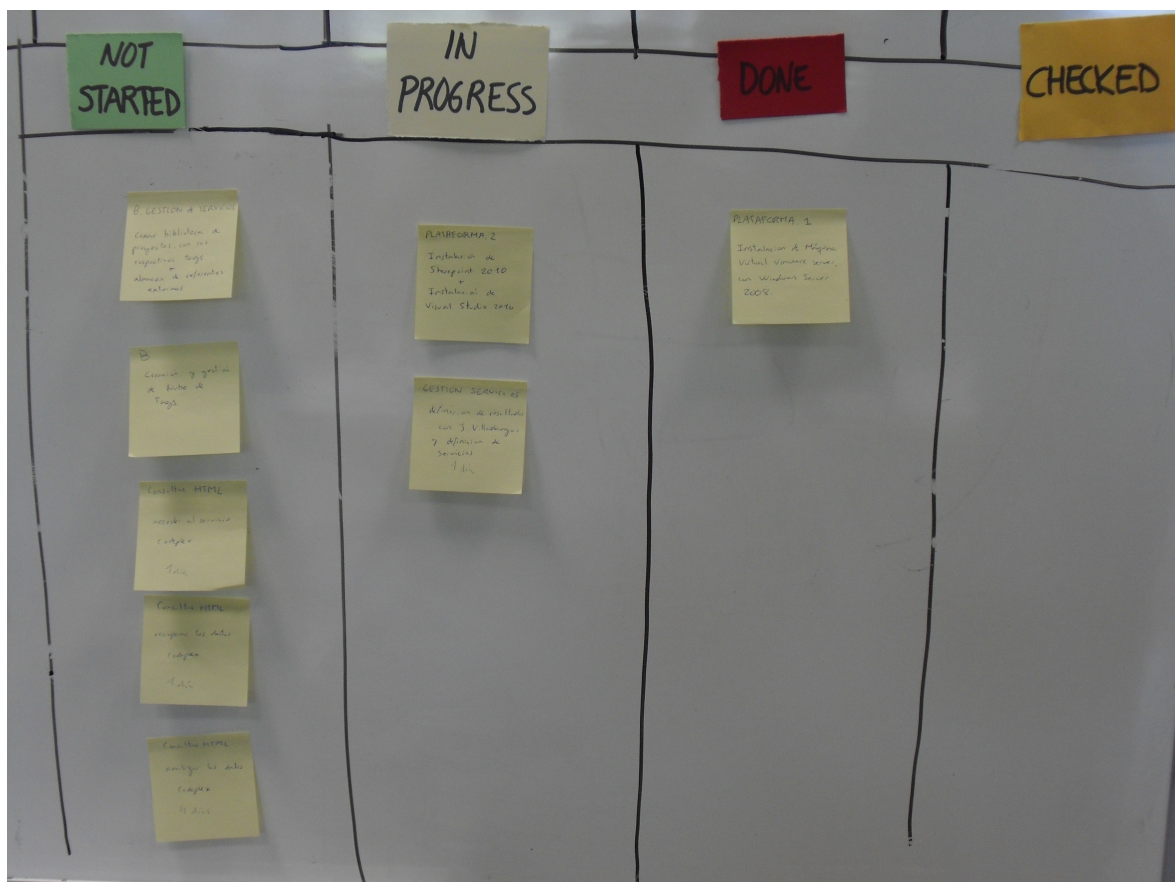


Figura 5.2 pizarra al inicio de Sprint 1 parte dcha

## 5.2 Seguimiento de las tareas

Una herramienta muy útil para el seguimiento del desarrollo de un proyecto es el uso de representaciones gráficas. Teniendo en cuenta las estimaciones realizadas para las historias de usuario, en la figura 5.3, tenemos la duración de cada tarea en azul, y la estimación como puntos de color rojo. Por ejemplo para la historia Plataforma las tareas a realizar son la instalación de VMware Server por un lado y SharePoint 2010 por otro. En la gráfica se observa que ambas tareas tienen una duración de 15 horas siendo la estimación de esta historia 18. En este caso la estimación ha sido bastante ajustada.

Para la Gestión de servicios, dividida en tres tareas, la estimación se queda grande, ya que finalicé esta historia de usuario en la mitad de tiempo de lo estimado, 15 horas frente a 30 estimadas. Durante el desarrollo esto se verá compensado puesto que otras historias duraron más tiempo de lo estimado. Las estimaciones son sólo guías, es tremendamente difícil estimar con total precisión.

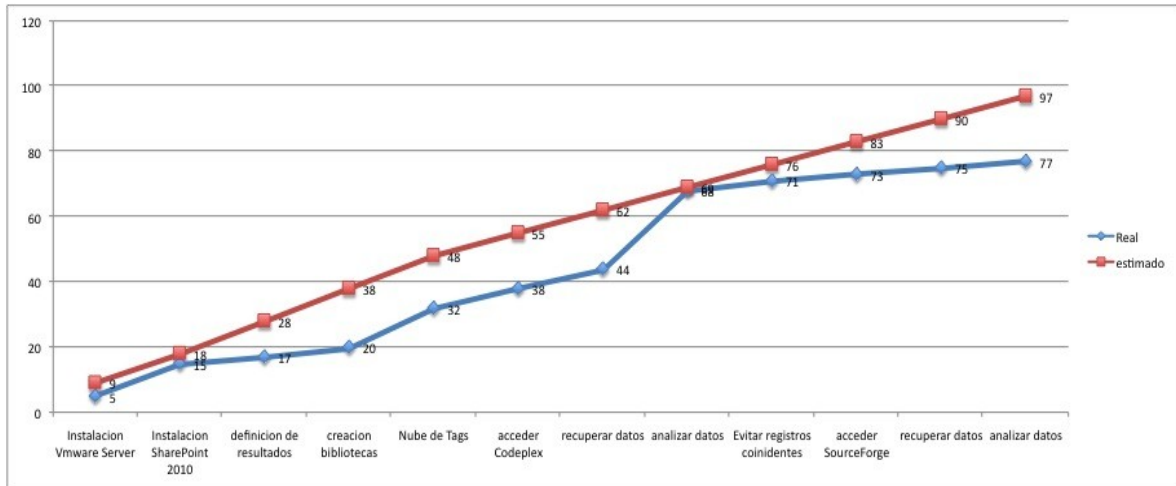
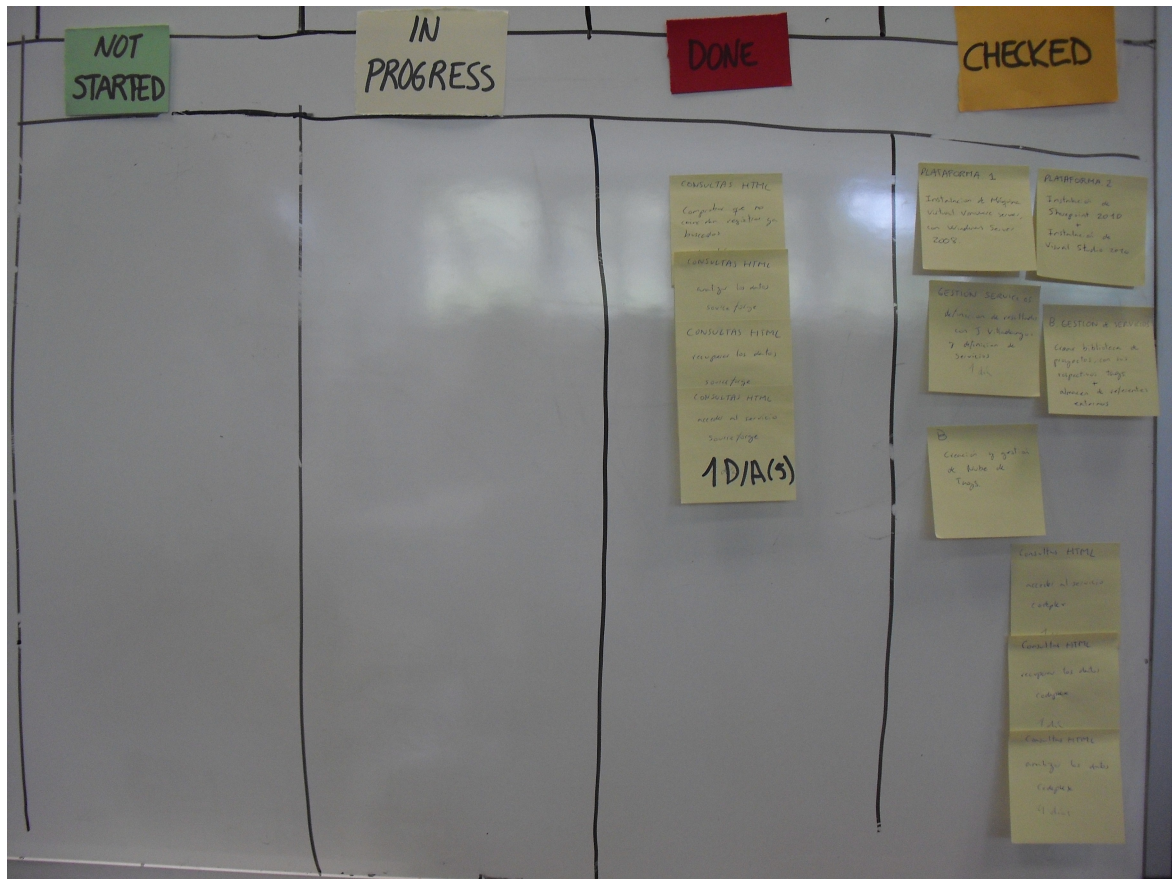


Figura 5.3 Gráfica Sprint 1

Como se observa en la gráfica el primer sprint fue un éxito, ya que también las consultas HTML se realizaron antes de lo estimado, esto se debe en parte a las pruebas de concepto realizadas previamente. Además las tareas de esta última historia eran acceder, recuperar y analizar los datos provenientes de una búsqueda en Codeplex, y lo mismo para SourceForge. Para luego registrarlos en el registro de mi sitio de SharePoint. Una vez hecho para una de las páginas fue muy similar hacerlo para la otra web, lo que hizo que el tiempo disminuyese.

La pizarra al final del sprint se ve en la fotografía de abajo, donde todas las tareas están bien en *checked* o en *done*. La diferencia entre ambos estados es que cuando finalizo una tarea la coloco en *done*, cuando mi compañera comprobaba que la tarea estaba satisfactoriamente resuelta la pasaba a *checked*. Al final del sprint las tareas que aun no habían sido comprobadas (*checked*) lo hacían en las reuniones de retrospectiva con el scrum master.





5.3 Foto de final de Sprint 1

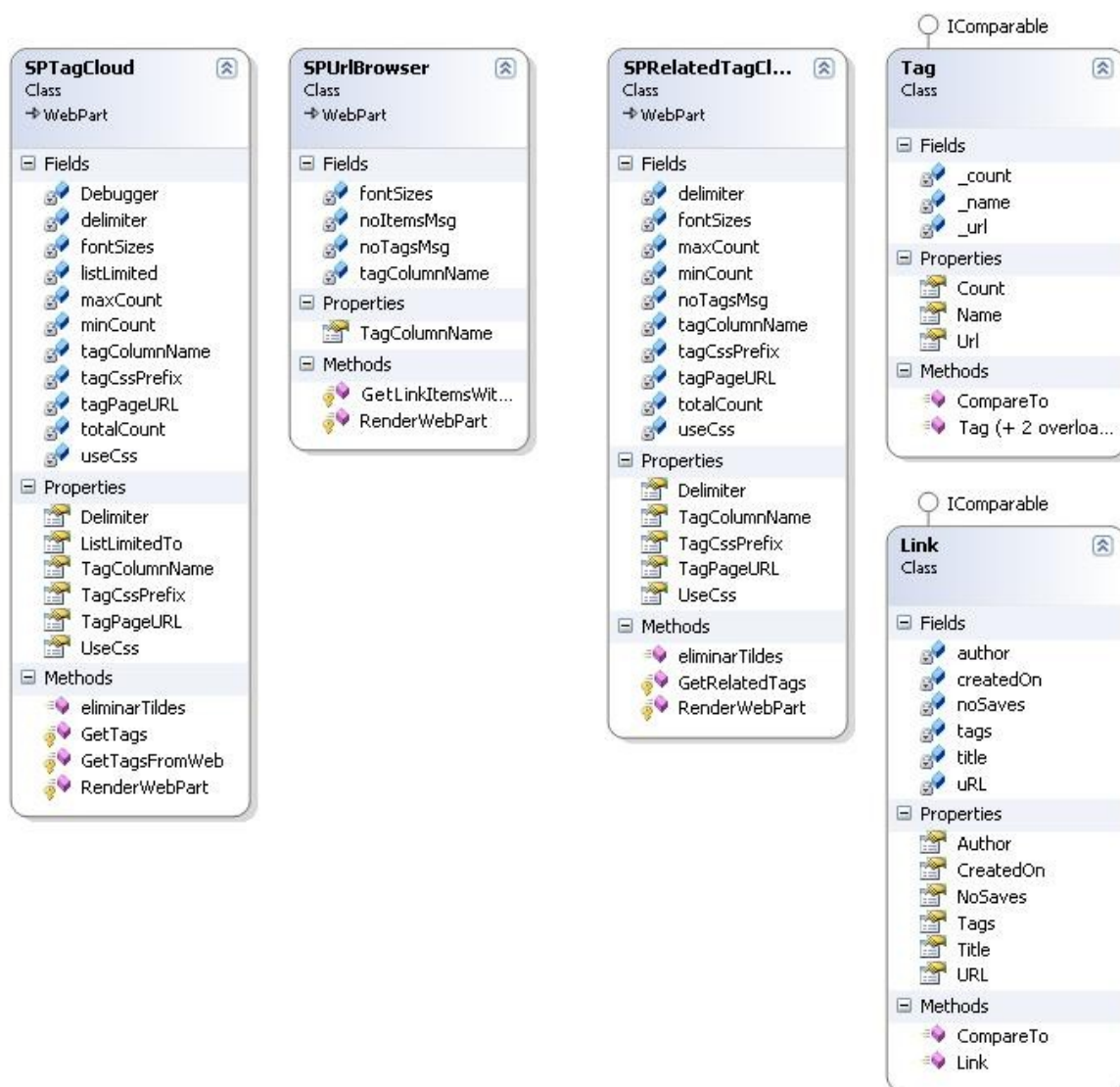
### 5.3 Diagrama de clases y listas creadas

En el desarrollo de este sprint implementé una funcionalidad primaria como eran las búsquedas HTML, los almacenes necesarios en SharePoint en Gestión de Servicios, y la navegación de resultados a través de la nube de Tags.

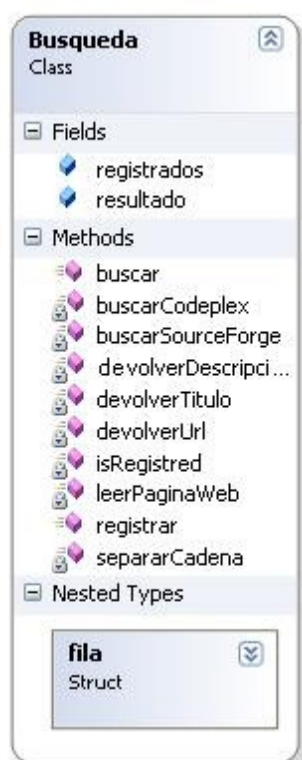
Para la nube de Tags hice uso de una disponible en Codeplex llamada CKS (Community Kit for SharePoint)[7] Tag Cloud WebPart, esta versión incluye varias WebParts para la navegación: la propia nube, una nube de tags relacionados, la WebPart de resultados y la WebPart de resultados a través de URL, para esta última se trata de indexar una o varias listas que posean además de etiquetas una columna llamada *URL* la cual servirá para mostrar los resultados .

De estas uso para mi proyecto las llamadas SPTagCloud (Nube) en la página principal, y en la página de resultados: SPUrlBrowser, para mostrar los proyectos con la etiqueta marcada en la nube, y SPRelatedTagCloud, donde muestro las etiquetas relacionadas con la que acabo de buscar. Los diagramas de clase de esta Nube de Tags de Codeplex se pueden ver en la figura 5.4.

Estas clases venían ya implementadas, más adelante tuve que modificar algunos detalles. Por ejemplo se puede ver en los métodos de SPRelatedTagCloud que añadí *eliminarTildes()*. Y es que la nube tenía algunos fallos como las tildes y las almohadillas (no las interpretaba bien) o las mayúsculas. Esto puede deberse a que la WebPart estaba ideada para una versión anterior de SharePoint.



5.4 Diagrama de clases de Nube de Tags



La clase que sí construí por completo fue la encargada de las búsquedas Html, cuyo funcionamiento ya expliqué en el capítulo de herramientas y pruebas de concepto. La clase creada la denominé búsquedas y la utilizaré y modificaré a lo largo de todo el desarrollo.

He creado la estructura *fila* donde almaceno cada referencia, resultado es una tabla de referencias por tanto de *filas*. En el campo registrados almaceno el número de referencias que se han registrado en el sitio de SharePoint.

Los métodos implementados permiten que el programa interprete que es lo que el usuario desea buscar, en que página (Codeplex o Sourceforge), y los resultados los registra si no estaban ya registrados, comprobándolo con el método *isRegistered()*

Figura 5.5 diagrama de clase búsqueda



En cuanto a almacenes y listas creé las siguientes:

- Biblioteca de proyectos: Una biblioteca en SharePoint es un tipo de lista donde podemos almacenar documentos de todo tipo y administrarlos en carpetas con columnas ya predefinidas. Aquí es donde se almacenan físicamente los proyectos subidos por el usuario, con su correspondiente código y documentación.
- Lista de proyectos: Desde la biblioteca, la gestión de los estados de los proyectos ( si están validados, sus etiquetas, los flujos de los tags..) se antojaban complejas debido a que es una lista predefinida. Así creamos una lista sencilla desde donde se gestionan los proyectos, en esta lista se almacena para cada proyecto: su título( igual que el de la carpeta del proyecto), un identificador único, un apuntador o enlace al proyecto en la biblioteca, sus etiquetas o tags, su estado y las columnas que usa el flujo para indicar los resultados de las búsquedas automáticas. Además de las columnas predefinidas, como por ejemplo, la fecha de creación, ultima modificación...
- Registro de búsquedas: La lista almacena las referencias provenientes de las búsquedas, para ello en la Gestión de servicios se definieron los datos y columnas que convenía almacenar para cada referencia. Esto incluye el Título, la descripción, la URL y los tags de la propia referencia. Además incluí una columna llamada Aprobación, que por defecto se inicia a Pendiente. Una vez que el usuario aprueba la referencia (cambiando Pendiente por Aprobado) se muestra en la navegación de resultados. Esta lista ha ido cambiando a lo largo del desarrollo, por ejemplo con las consultas XML añadí una columna que indica que motor de búsqueda se ha usado para la consulta.

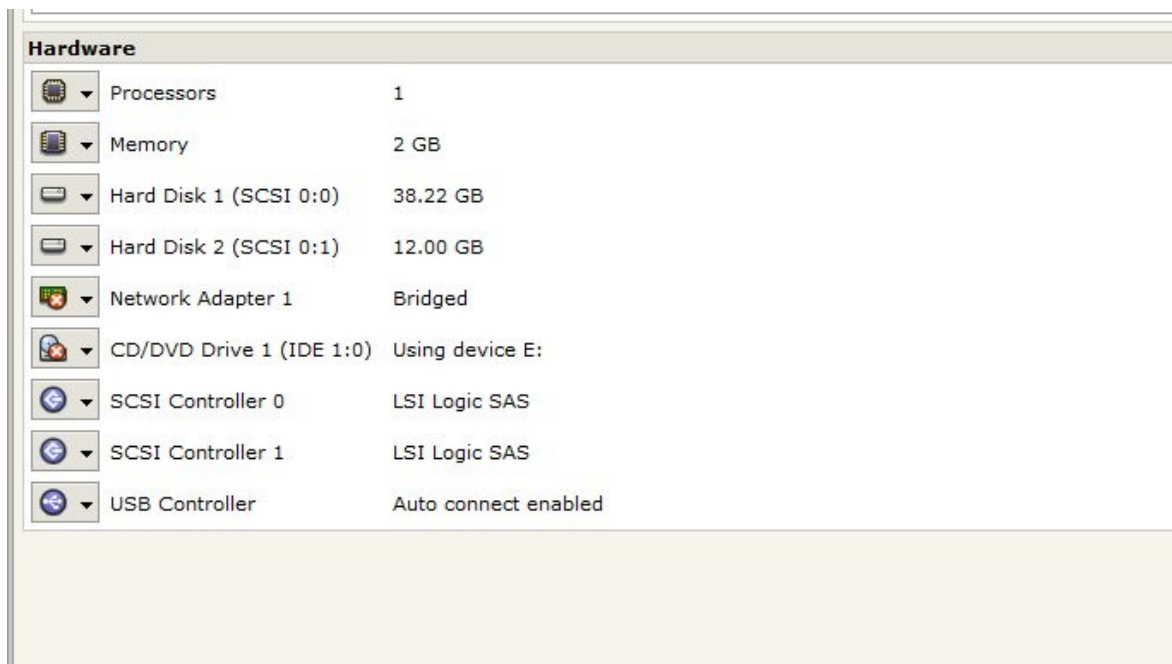
#### 5.4 Incidencias destacables

A la hora de instalar la plataforma, una de las incidencias que encontramos, fue que el nuevo SharePoint sólo era compatible con un sistema operativo de 64 bits. En ese momento contábamos con Windows 7 de 32 con una máquina virtual Windows Virtual PC, tuvimos pues que cambiar nuestro sistema operativo a Windows 7 de 64, de esta manera también pudimos aprovechar mejor la RAM de la que disponíamos. La máquina virtual que mejor se adaptaba a nuestras necesidades resultó ser VMware Server .

Los requisitos hardware iniciales de SharePoint 2010 indicados por Microsoft se pueden ver en la siguiente tabla:

Componente	Requisito mínimo
Procesador	64 bits, cuatro núcleos
RAM	4 GB para evaluación o uso del programador 8 GB para la instalación de granjas con varios servidores y un solo servidor para uso de producción
Disco duro	80 GB para la unidad de sistema Para el uso de producción, se necesita espacio libre en disco adicional para las operaciones cotidianas. Mantenga el doble de espacio libre en comparación con la memoria RAM para entornos de producción.

No obstante, nuestra máquina no cumplía todos estos requisitos, ya que no contábamos con un procesador de cuatro núcleos, nuestra RAM asignada a VMware era de 2 GB y almacenamiento de 50GB (creé dos discos duros, porque el primero se quedó corto). Esto influyó en la rapidez de SharePoint y nos hizo imposible restaurar un backup de otro sitio puesto que no había espacio, pero pudimos desarrollar nuestra aplicación web en un sitio creado sin problemas. La configuración hardware extraída del propio VMware Server se puede observar en la captura de abajo.



Captura 5.6 hardware de la máquina

## 5.5 Producto entregable

Como se vio en el capítulo referente a las metodologías ágiles, Scrum se caracteriza por entregas frecuentes de producto funcional al usuario. Esto consigue maximizar el valor del esfuerzo realizado, además de involucrar más al cliente. Así pues, también en este desarrollo se hicieron retrospectivas donde al cliente, en este caso al coordinador del proyecto, se le mostraba la funcionalidad desarrollada hasta el momento.

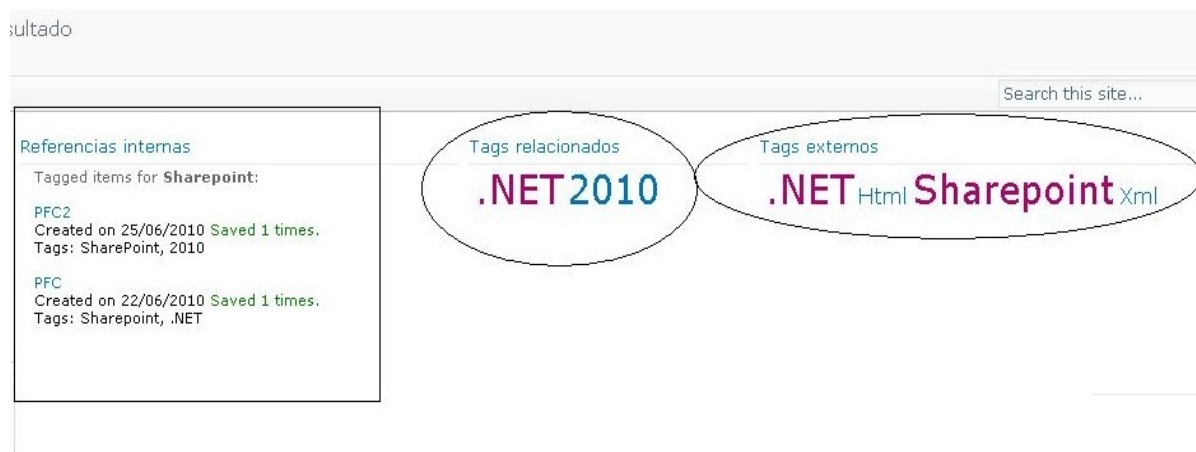
Para este sprint las funcionalidades implementadas y finalizadas son las siguientes:

- Buscar proyectos a través de la nube de etiquetas: en la siguiente página podemos ver la página principal, donde se encuentra la nube de tags, y la de resultados. Cada etiqueta de la nube es un vínculo a la página de resultados para ese tag. Indicado mediante una *query* en la Url.

En la captura 5.8 está la página de resultados para el tag *SharePoint*, en ella he remarcado los elementos web con un cuadrado y dos círculos. En el cuadrado se encuentran los proyectos que tienen a *SharePoint* entre sus etiquetas. Esta WebPart es SPUrlBrowser.



Captura 5.7 nube de tags









Captura 5.8 resultados para SharePoint

En el círculo de su izquierda se encuentra la WebPart de tags relacionados. En ella vemos las etiquetas que comparten los proyectos internos con la etiqueta para la que hemos buscado. Así en este ejemplo, para *SharePoint* los tags relacionados serían *.NET* y *2010*, esta se comporta igual que la nube de tags de la página principal, redireccionando a la misma página con la nueva etiqueta marcada. Por último la otra nube redondeada es la nube principal que englobará los proyectos internos y externos.

- La siguiente y última funcionalidad desarrollada en este sprint fueron las consultas Html. Que aunque aún no estaban disponibles en la web, si qué desde Visual Studio se registraban en SharePoint las referencias procedentes de una búsqueda. Podemos ver el registro de búsquedas en la siguiente página.

miSitio ▶ RegistroBusqueda : Vista del Registro ▼

Registro de las búsquedas de proyectos externos

Icono	Título	descripcion	URL
	XAware Data Integration Project	Create services to integrate applications & move data of any type. Build data &quot;views&quot; across DBMS, SOAP, HTTP/REST, Salesforce, SAP, Microsoft, SharePoint, Text, LDAP, FTP sources to read, write & transfer data. Eclipse designer & runtime engine.	<a href="http://www.sourceforge.net/projects/xaware/">www.sourceforge.net/projects/xaware/</a>
	UbuntuZilla: Mozilla Software Installer	An APT repository hosting the Mozilla builds of the latest official releases of Firefox, Thunderbird, and Seamonkey. Do not manually download the files - read the website, <a href="http://ubuntuzilla.sourceforge.net">http://ubuntuzilla.sourceforge.net</a> for instructions on using the repository.	<a href="http://www.sourceforge.net/projects/ubuntuzilla/">www.sourceforge.net/projects/ubuntuzilla/</a>
	TripleA Maps	A project by Vegryn for creating, centralizing, and hosting all open-source TripleA maps, as well as other developer resources such as artwork and map makers. If you need the actual TripleA program, go here: <a href="https://sourceforge.net/projects/triplea/">https://sourceforge.net/projects/triplea/</a>	<a href="http://www.sourceforge.net/projects/tripleamaps/">www.sourceforge.net/projects/tripleamaps/</a>
	The Tao Framework	The Tao Framework for .NET is a collection of bindings to facilitate cross-platform game-related development utilizing the .NET platform.	<a href="http://www.sourceforge.net/projects/taoframework/">www.sourceforge.net/projects/taoframework/</a>
	StarDict	StarDict is a Cross-Platform and international dictionary written in Gtk2. It has powerful features such as &quot;Glob-style pattern matching&quot;, &quot;Scan selection word&quot;, &quot;Fuzzy search&quot;, &quot;Net dict&quot;, &quot;Full text translation&quot; etc.	<a href="http://www.sourceforge.net/projects/stardict/">www.sourceforge.net/projects/stardict/</a>
	ADO.NET 2.0 Provider for SQLite	ADO.NET 2.0/3.5 Provider for SQLite. Supports Full and Compact Framework, Entity Framework, complete with full Visual Studio 2005/2008 design-time	<a href="http://www.sourceforge.net/projects/sqlite-dotnet2/">www.sourceforge.net/projects/sqlite-dotnet2/</a>

Captura 5.9 Registro de búsquedas



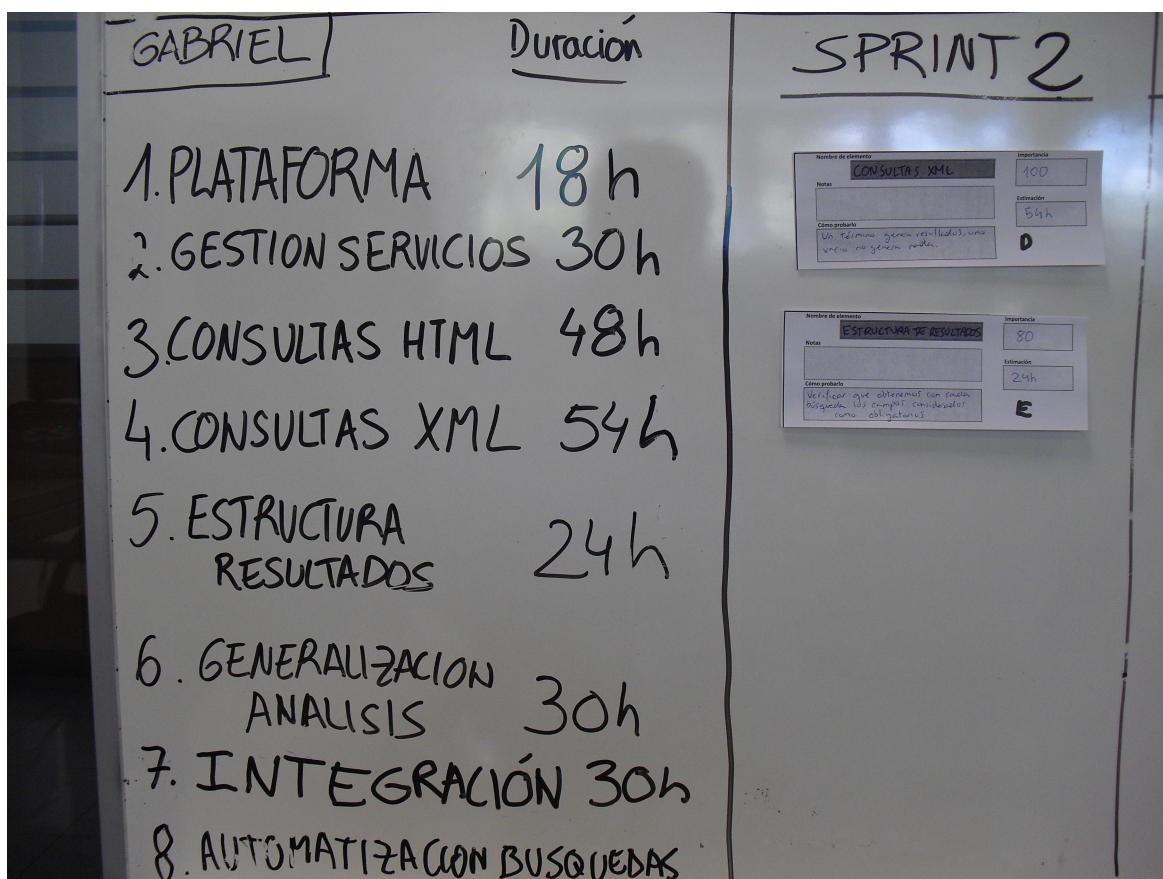
## 6. Desarrollo: Sprint 2

### 6.1 Selección de requisitos

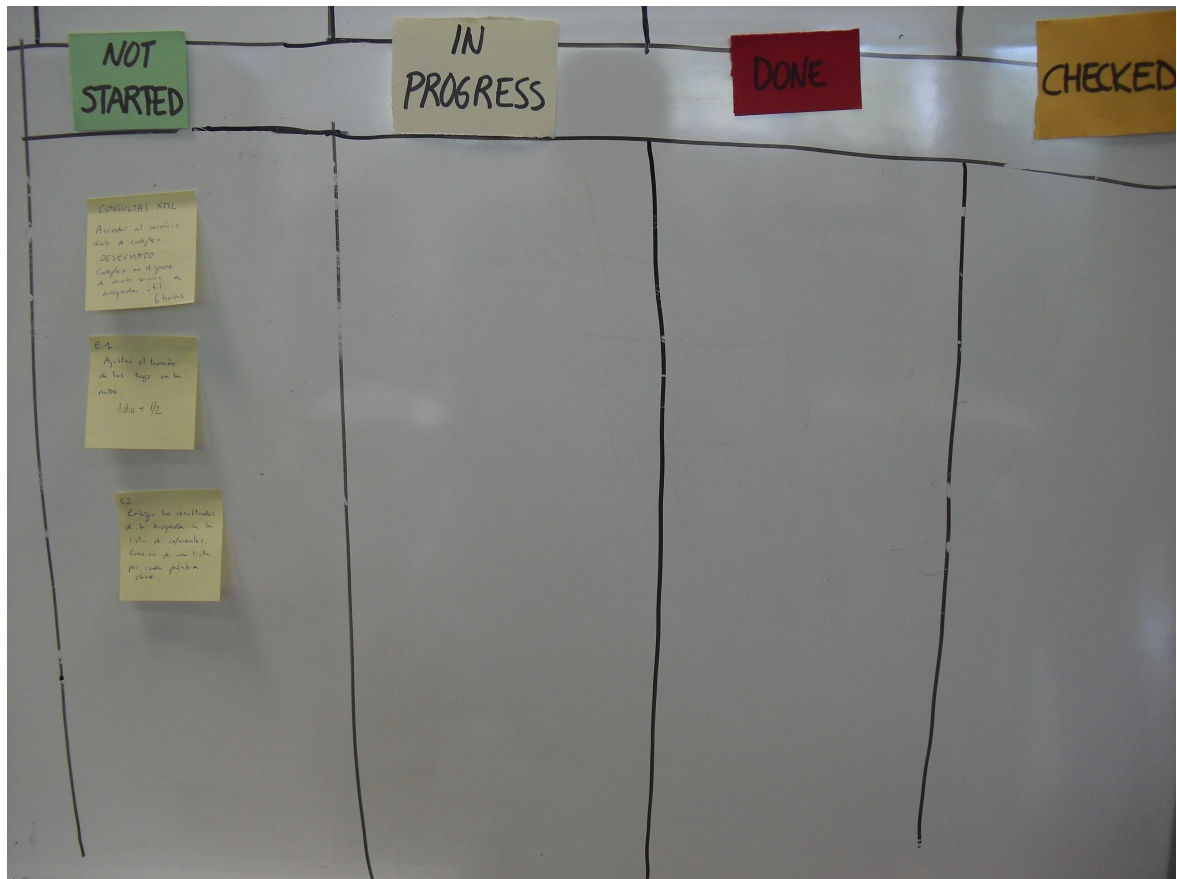
Las historias de usuario del sprint anterior se acabaron a tiempo, por lo que para el nuevo sprint decidimos seguir con las prioridades establecidas en el product backlog e incluir las historias de usuario más prioritarias. La automatización de búsquedas es una de las historias con más importancia, sin embargo no se puede automatizar algo que no está implementado.

Por tanto las siguientes historias a realizar resultaron las consultas Xml y la estructura de resultados. En esta última se debía definir como mostrar los referentes encontrados al usuario cuando navegase por la nube, además decidimos incluir la posibilidad de que el usuario aprobase o no una referencia, sólo se le mostrarían entonces las referencias aprobadas.

Las consultas Xml ya las definí en el capítulo tres, la idea principal era usar web services de búsqueda de páginas concretas. Gran parte de lo realizado en este sprint se desechó en las reuniones posteriores, se puede decir que fue el sprint menos productivo, aunque aquí fue donde mejor se definió lo que realmente se quería que fuese la aplicación. Más adelante en el capítulo de incidencias explicaré esto con más detalle.



Fotografía 6.1 Pizarra de inicio de Sprint 2 parte izda.



Fotografía 6.2 Pizarra al inicio de Sprint 2 parte dcha

## 6.2 Seguimiento del proyecto

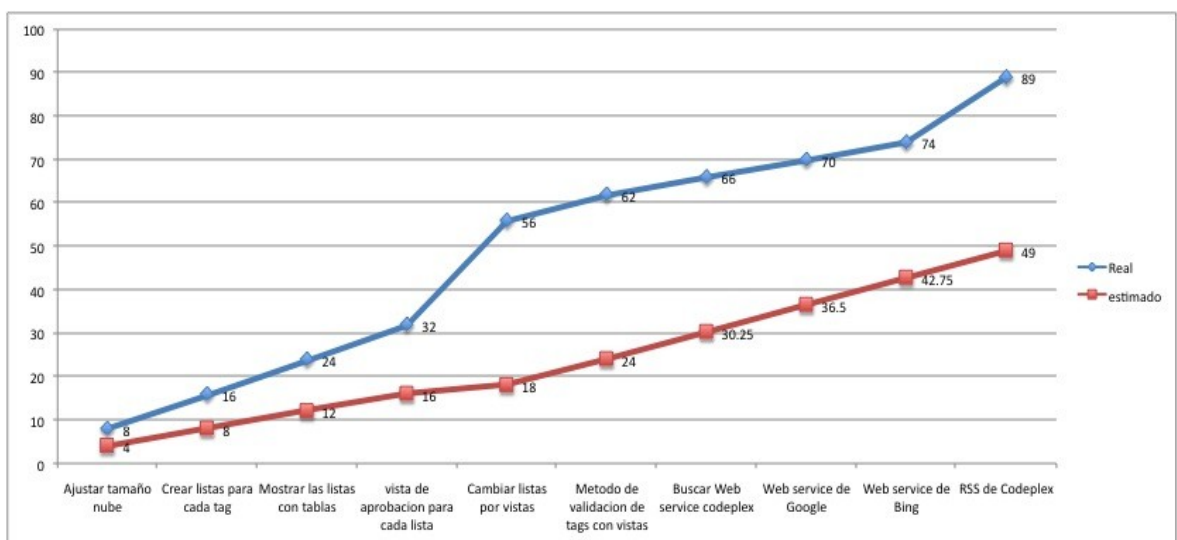


Figura 6.3 Gráfica de Sprint 2

La gráfica muestra claramente como la linea de lo estimado se encuentra muy por debajo de la real, como ya he dicho en este sprint se desechó la primera aplicación que creé para la historia de usuario: estructura de resultados. La primer idea que tuve fue mostrar los resultados a través de tablas. Estas no son las tablas *arrays* típicas de almacenamiento de datos, sino controles gráficos de muestra de datos en la web usando Html. Para rellenar



estas tablas creaba una lista oculta de SharePoint para cada etiqueta. De este modo cuando el usuario buscaba referencias para una etiqueta concreta, la tabla se rellenaría con el contenido de la lista de dicha referencia.

Esta implementación corresponde a las tareas: *crear lista para cada tag, mostrar los resultados con tablas y crear vistas de aprobación para cada lista*. En las reuniones semanales correspondientes vimos que esta solución no era la más indicada, puesto que se repetía información cuando una referencia posee varias etiquetas, y además es poco eficiente. Se propuso otra forma basada en vistas (SPView en SharePoint) a una única lista, la lista *Registro búsquedas*. Estas son las siguientes tareas desarrolladas; *cambiar el sistema de listas por el de vistas y crear un método de validación*, es decir, que el usuario apruebe las referencias.

Para realizar todo esto me ayudé de una nueva clase que aporta el framework de SharePoint 2010 para Visual Studio 2010, estas son las *XsltListViewWebPart*. Esta nueva clase permitió el uso de una única lista, dando dinamismo a la aplicación.

Una vez cambiado el sistema desechado anteriormente, estaba pendiente refinar la estructura en algunos aspectos, por lo que se mantendría esta historia en el siguiente sprint. Para continuar, pasé a las consultas Xml. En la gráfica se puede apreciar que el punto de esta historia de usuario está fuera de la gráfica, con esto quiero indicar que no se completó. Es más, muchas de las tareas que aparecen allí de web services de Bing o Google, o leer los RSS de Codeplex fueron desechadas o se encontraban obsoletas (como el servicio SOAP de Google).

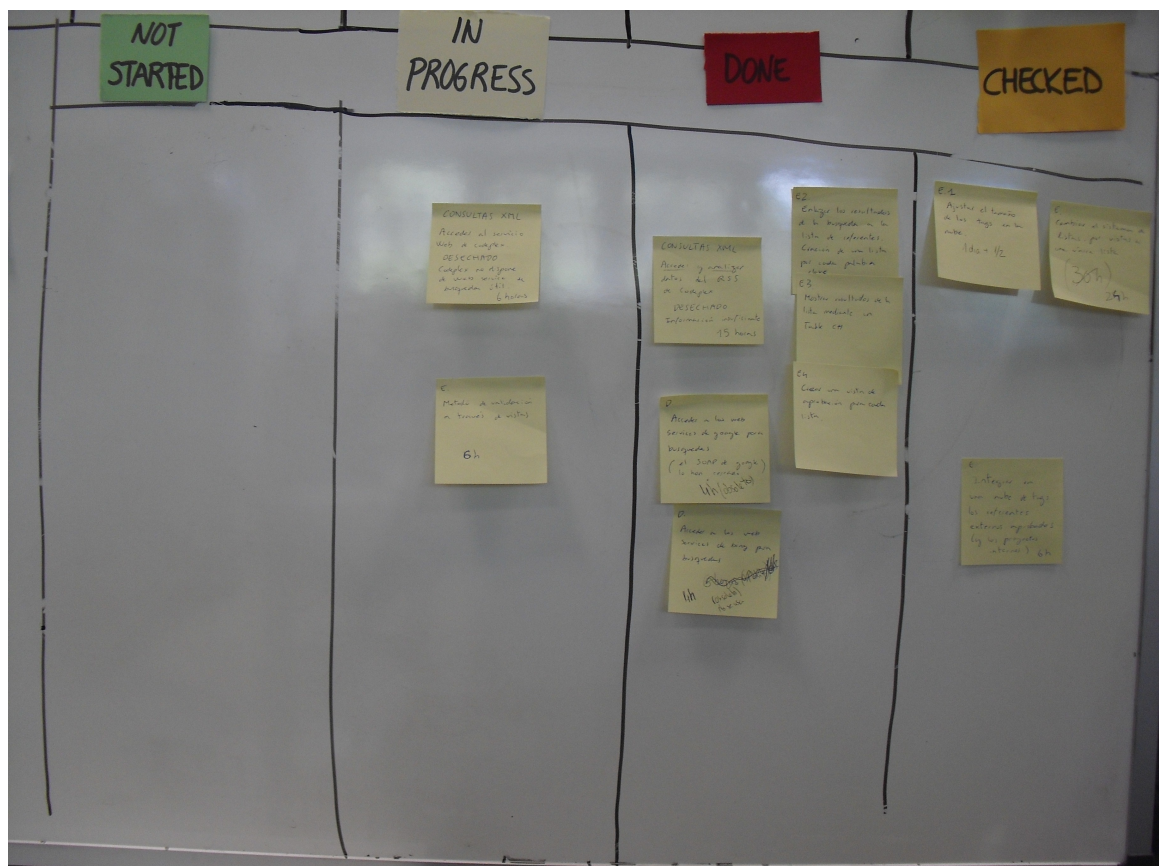


Figura 6.4 pizarra al final del sprint 2

### 6.3 Diagramas de clases y listas

De las clases que desarrollé en este sprint la que finalmente sería útil en el proyecto fue la WebPart a la que llamé *ExternViewsWebPart*. Esta WebPart se instala en la página de resultados. Muestra los referentes externos aprobados por el usuario junto a un botón que permite aprobar más referentes al hacer click, desplegando una vista de enlaces pendientes.

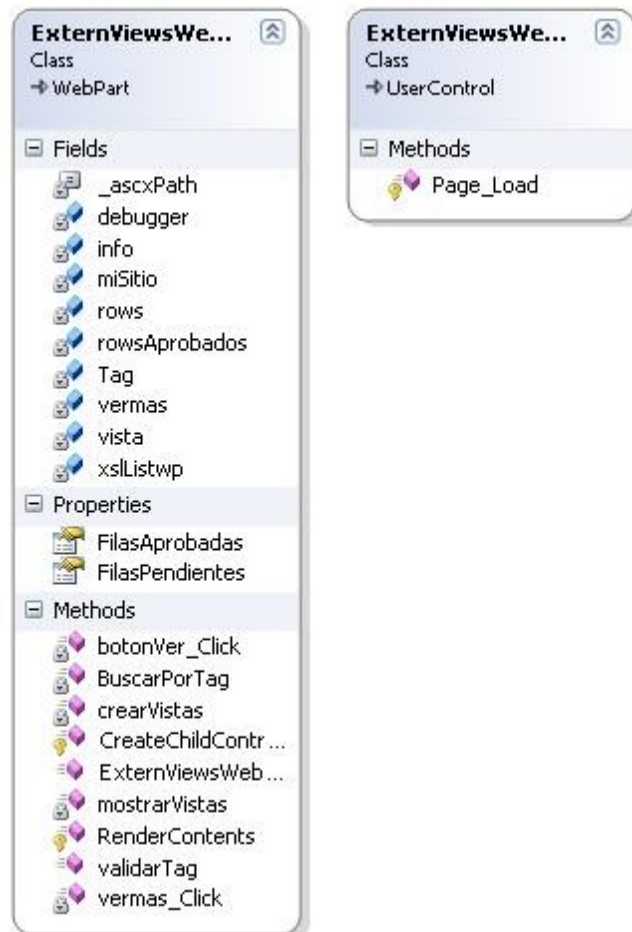


Figura 6.5 Diagrama de clase

En el diagrama podemos observar que esta clase contiene entre otros campos el campo *xslListwp* perteneciente a la clase *XsltListViewWebPart*, en ella cargo las dos vistas que usaré para cada tag, la vista de referentes aprobados y la que incluye todos. Esta WebPart se instala en la misma página que la WebPart de la nube de tags: *SPURLBrowser*, donde se muestran los resultados de los proyectos subidos a la plataforma. De este modo al navegar por los proyectos de la plataforma veríamos referencias externas asociadas a estos, a través de la WebPart que he creado.

La nube de tags recurre a la URL para mandar la información de la etiqueta a buscar, desde la página que contiene la nube, a la página de resultados. Del mismo modo, muestro yo las referencias externas aprobadas. A través de la Url obtengo el tag, lo valido con el método *validarTag()*, comprobando que exista al menos una referencia (aprobada o pendiente). Si es la primera vez que el usuario busca esa etiqueta crearé las vistas de aprobados y pendientes para ella, en el método *crearVistas*. Si no mostraré directamente la vista de referentes aprobados con el método *mostrarVistas*. Esto se verá más claramente en el apartado de producto entregable al final del Sprint, mediante capturas de pantalla.



No se creó ninguna lista de SharePoint finalmente para este Sprint. En un primer planteamiento mi WebPart creaba listas ocultas, de forma dinámica, para cada referencia. Sin embargo, como ya he explicado, se optó por generar vistas a una única lista principal. Las vistas creadas para cada etiqueta eran:

- Vista de *<etiqueta>*: esta vista contiene todas las referencias o enlaces que contienen la etiqueta que lleva por nombre, sin importar el valor que tome la columna aprobación (Pendiente o Aprobado).
- Vista de *<etiqueta>* aprobados: esta vista contiene todas las referencias o enlaces que contienen la etiqueta que lleva por nombre, siempre que el valor de la columna aprobación sea *Aprobado*.

Ambas se generan de forma automática, y más adelante con la implementación del flujo también se eliminarían al eliminar la etiqueta que representan.

#### 6.4 Incidencias destacables


Las consultas Xml estuvieron marcadas por las incidencias. Cada tarea, representada en la gráfica, de esta historia de usuario fue desechada posteriormente. En primer lugar intenté buscar un servicio web de búsqueda para las páginas Codeplex y SourceForge, ante la falta de información, me puse en contacto con Codeplex que me afirmó que no disponían de un web service público. Durante las reuniones planteábamos diferentes maneras de realizar estas búsquedas personalizadas. Creé un lector de RSS para extraer los Feeds que estas páginas colgaban, sin embargo la información era insuficiente.

La solución final fue hacer uso de los grandes buscadores de la red para obtener la información de páginas concretas. Ésto iba a aportar mayor dinamismo a la aplicación ya que no me limitó a proponer una serie de webs, sino que es el usuario quien decide donde buscar. Comencé entonces a probar diferentes servicios web. El SOAP de Google se encontraba obsoleto, así que desarrollé una aplicación que consultaba usando el web service del buscador Bing. Este último funcionaba bien, sin embargo como ya expliqué en el capítulo de herramientas y pruebas de concepto, al tratar de subirlo al sitio de SharePoint fallaba en temas de permisos.

El final de este Sprint se vio marcado por estimaciones que no se cumplieron en el caso de la estructura de resultados, lo que conllevó a que la historia de usuario consultas Xml no pudiese finalizarse, se incluiría entonces en el siguiente Sprint.

#### 6.5 Producto entregable

La funcionalidad resultante en este Sprint está muy centrada en el usuario, pues es cuando realmente puede interactuar con la aplicación; bien sea aprobando referencias o buscándolas para su proyecto alojado en la plataforma.


**miSitio** ▶ resultado

Inicio

Documentos  
 Listas  
 Registro de Búsquedas  
 Proyectos  
 Biblioteca de Proyectos  
 Lista de Proyectos  
 Papelera de reciclaje  
 Todo el contenido del sitio

**Referencias internas**  
 Tagged items for **Sharepoint**:  
 PFC2  
 Created on 25/06/2010 Saved 1 times.  
 Tags: SharePoint, 2010  
 PFC  
 Created on 22/06/2010 Saved 1 times.  
 Tags: Sharepoint, .NET

**Tags relacionados**  
 .NET 2010

**Tags exte**  
 .NET

**Enlaces Externos**  
 Referencias encontradas para: **Sharepoint**  
 Mostrando vista de referentes aprobados

Título	URL	descripcion
iText.NET	<a href="http://www.sourceforge.net/projects/itextdotnet/">www.sourceforge.net/projects/itextdotnet/</a>	iText.NET is iText porting on .NET Framework by J#. This library allows you to generate PDF, (X)HTML, XML, RTF files on Microsoft.NET Framework including ASP.NET.
HP Linux Imaging and Printing	<a href="http://www.sourceforge.net/projects/hplip/">www.sourceforge.net/projects/hplip/</a>	The Hewlett-Packard Linux Imaging and Printing Project provides a unified single and multi-function connectivity solution for Linux. For support and help, please visit <a href="https://launchpad.net/hplip">https://launchpad.net/hplip</a>
Your Development Resource - CodeProject	<a href="http://www.codeproject.com/">http://www.codeproject.com/</a>	Integrating Sharepoint 2010 and SQL Reporting Services 2008 in 6 easy steps [Technical Blog] Raymund Macaalay [unedited] 17 Jun 2010 (new) Observer in .NET 4.0 with IObservable(T ...
Extended Interface for Toolbars - CodeProject	<a href="http://www.codeproject.com/KB/menus/extendedtoolbar.aspx">http://www.codeproject.com/KB/menus/extendedtoolbar.aspx</a>	Select the tab .NET Framework Components; Click B rowse... Click on ExtendedInterface.DLL; Click OK. The Toolbox should have two new components: StatusMessage and ToolbarFunction.

Aprobar/Editar referencias

Captura 6.6 Enlaces externos

La vista de las referencias aprobadas para una etiqueta solicitada aparece justo debajo de los proyectos internos. Esta vista es una `XsltListViewWebPart` que permite al usuario trabajar con ella como si de una lista cualquiera se tratará; puede ordenar las referencias en función de el título, Url o descripción. Además añadí un botón debajo de la vista que al presionarlo , y tal y como anuncia su nombre, se nos desplegarán una serie de referencias que podremos aprobar o editar. Esta nueva vista contiene más columnas una de ellas *Aprobación* que admite los valores *Pendiente* y *Aprobado*.

Titulo

URL

descripcion

iText.NET
 [www.sourceforge.net/projects/itextdotnet/](http://www.sourceforge.net/projects/itextdotnet/)
 iText.NET is iText porting on .NET Framework by J#. This library allows you to generate PDF, (X)HTML, XML, RTF files on Microsoft.NET Framework including ASP.NET.

HP Linux Imaging and Printing
 [www.sourceforge.net/projects/hplip/](http://www.sourceforge.net/projects/hplip/)
 The Hewlett-Packard Linux Imaging and Printing Project provides a unified single and multi-function connectivity solution for Linux. For support and help, please visit <https://launchpad.net/hplip>

Your Development Resource - CodeProject
 <http://www.codeproject.com/>
 Integrating Sharepoint 2010 and SQL Reporting Services 2008 in 6 easy steps [Technical Blog] Raymund Macaalay [unedited] 17 Jun 2010 (new) Observer in .NET 4.0 with IObservable(T ...

Extended Interface for Toolbars - CodeProject
 <http://www.codeproject.com/KB/menus/extendedtoolbar.aspx>
 Select the tab .NET Framework Components; Click B rowse... Click on ExtendedInterface.DLL; Click OK. The Toolbox should have two new components: StatusMessage and ToolbarFunction.

Aprobar/Editar referencias

Seleccione los que desee aprobar y pulse el boton

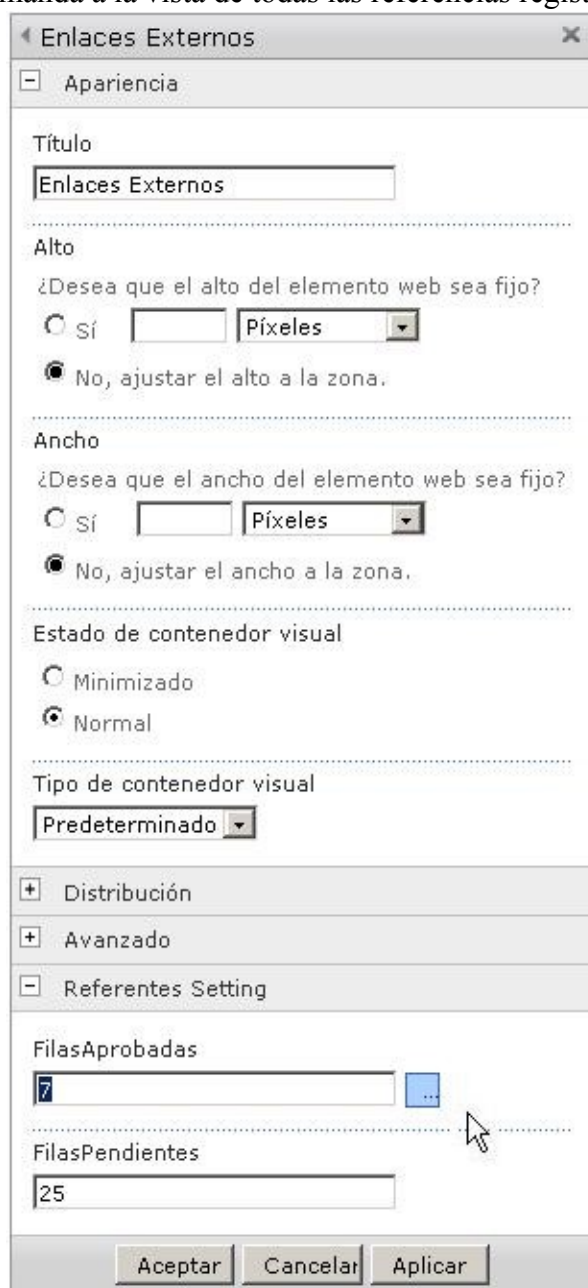
Título	URL	descripcion	posibles tags	Aprobacion
Ubuntuzilla: Mozilla Software Installer	<a href="http://www.sourceforge.net/projects/ubuntuzilla/">www.sourceforge.net/projects/ubuntuzilla/</a>	An APT repository hosting the Mozilla builds of the latest official releases of Firefox, Thunderbird, and Seamonkey. Do not manually download the files - read the website, <a href="http://ubuntuzilla.sourceforge.net">http://ubuntuzilla.sourceforge.net</a> for instructions on using the repository.	Sharepoint, .NET	Pendiente
net-snmp	<a href="http://www.sourceforge.net/projects/net-snmp/">www.sourceforge.net/projects/net-snmp/</a>	net-snmp provides tools and libraries relating to the Simple Network	Sharepoint, .NET	Pendiente
emule Xtreme Mod	<a href="http://www.sourceforge.net/projects/emulextreme/">www.sourceforge.net/projects/emulextreme/</a>	emule client, based on official emule version (see <a href="http://www.emule-project.net">www.emule-project.net</a> )	Sharepoint, .NET	Pendiente
StarDict	<a href="http://www.sourceforge.net/projects/stardict/">www.sourceforge.net/projects/stardict/</a>	StarDict is a Cross-Platform and international dictionary written in C++2. It has powerful features such as &quot;Glob-style pattern matching&quot;, &quot;Scan selection word&quot;, &quot;Fuzzy search&quot;, &quot;Net dict&quot;, &quot;Full text translation&quot;, etc.	Sharepoint, .NET	Pendiente
SharpDevelop	<a href="http://www.sourceforge.net/projects/sharppdevelop/">www.sourceforge.net/projects/sharppdevelop/</a>	SharpDevelop is an open source IDE for the .NET platform. It is entirely written in C#, and supports various project types in multiple programming languages.	Sharepoint, .NET	Pendiente
SharePoint List Copy	<a href="http://www.sourceforge.net/projects/splistcp/">www.sourceforge.net/projects/splistcp/</a>	A clean and reliable way to migrate from SPS 2003 to MOSS 2007. This is a .NET/C# command line program that copies all your SharePoint 2003 lists and document libraries to a SharePoint 2007 server, keeping metadata like modification date.	Sharepoint, .NET	Pendiente
iText.NET	<a href="http://www.sourceforge.net/projects/itextdotnet/">www.sourceforge.net/projects/itextdotnet/</a>	iText.NET is iText porting on .NET Framework by J#. This library allows you to generate PDF, (X)HTML, XML, RTF files on Microsoft.NET Framework including ASP.NET.	Sharepoint, .NET	Aprobado
FCKeditor	<a href="http://www.sourceforge.net/projects/fckeditor/">www.sourceforge.net/projects/fckeditor/</a>	Online text editor (DHTML editor), for ASP, ASP.NET, ColdFusion, PHP, Java and JavaScript brings to the web many of the powerful features of known desktop editors like Word. It&apos;s XHTML compliant and works with Firefox, Mozilla,	Sharepoint, .NET	Pendiente

Captura 6.7 referencias por aprobar



Captura 6.8 aprobación de una referencia

Cuando pulsamos el botón de guardar la referencia se verá en la vista de referencias aprobadas para *SharePoint* y también para *.NET*, puesto que es la otra etiqueta. El número de referencias de las vistas el usuario lo puede editar desde el editor gráfico de la WebPart. El editor se puede ver en la captura 6.9, el personalizado está en la sección *Referentes Settings*. Además dispone al final de la página de resultados un enlace llamado *ver todas* que le manda a la vista de todas las referencias registradas para ese tag.



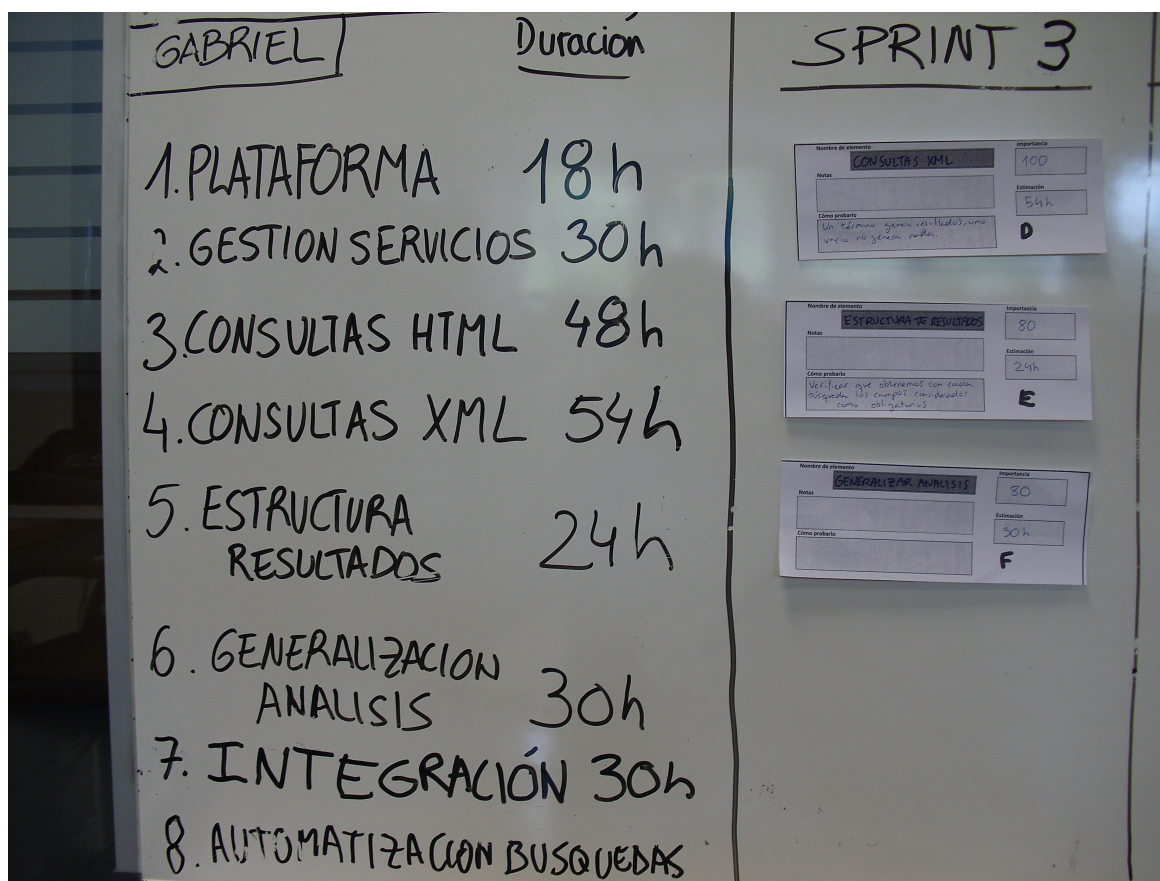
Captura 6.9 Editor de WebPart

## 7. Desarrollo: Sprint 3

### 7.1 Selección de requisitos

En vista de que en el pasado sprint las consultas Xml tan sólo se iniciaron y que estaba pendiente un refinamiento de la estructura de resultados, se mantuvieron estas historias de usuario en este sprint para que fueran finalizadas.

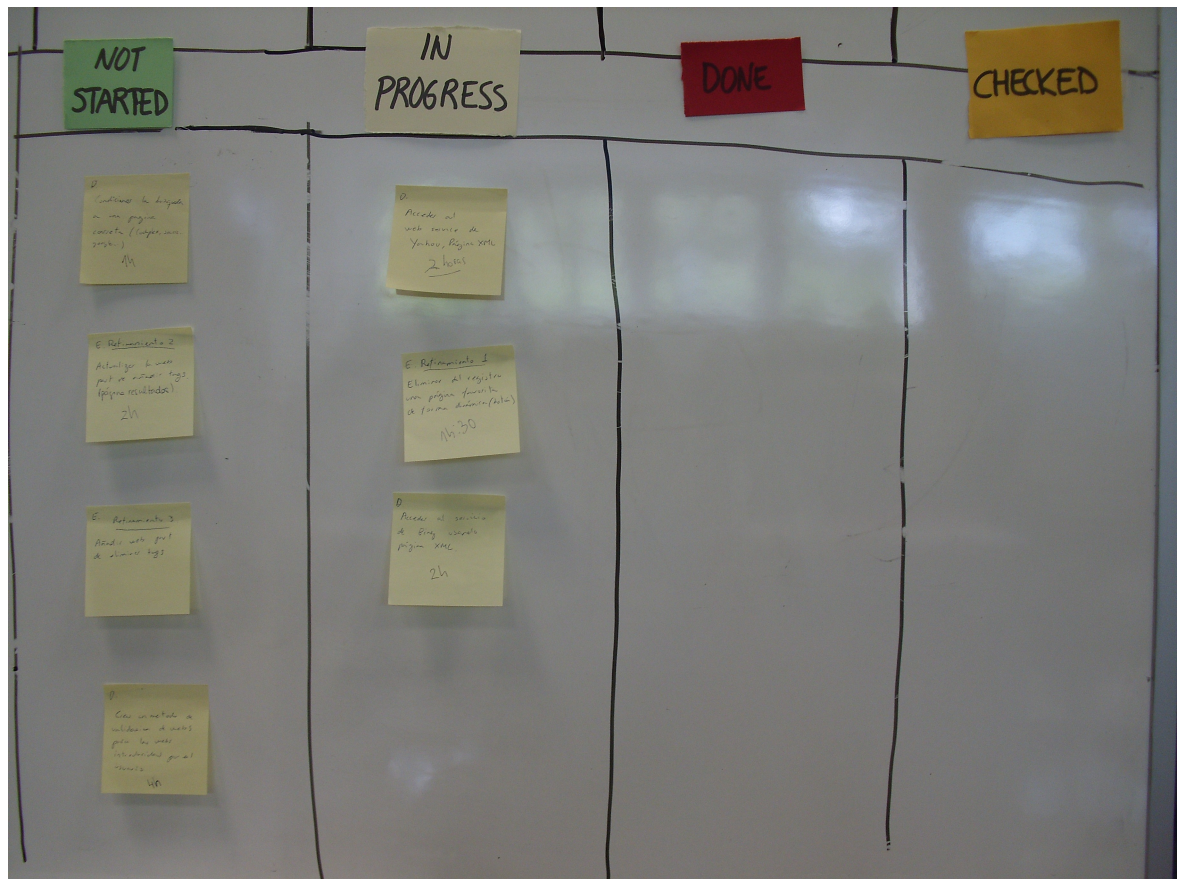
Además se añadió la historia de usuario Generalización y análisis, que consistía básicamente en integrar todo lo desarrollado. Las búsquedas Html con las Xml, haciendo al usuario la gestión de estas independiente de los resultados que obtiene. Además la generalización incluye la posibilidad de añadir webs donde buscar y que estas se almacenen. Al final de este sprint en el producto entregable el usuario será capaz de realizar consultas de referentes externos de forma manual, seleccionando las páginas donde desea buscar, y utilizando el motor de búsquedas que considere más apropiado.



Fotografía 7.1 pizarra al inicio de Sprint 3

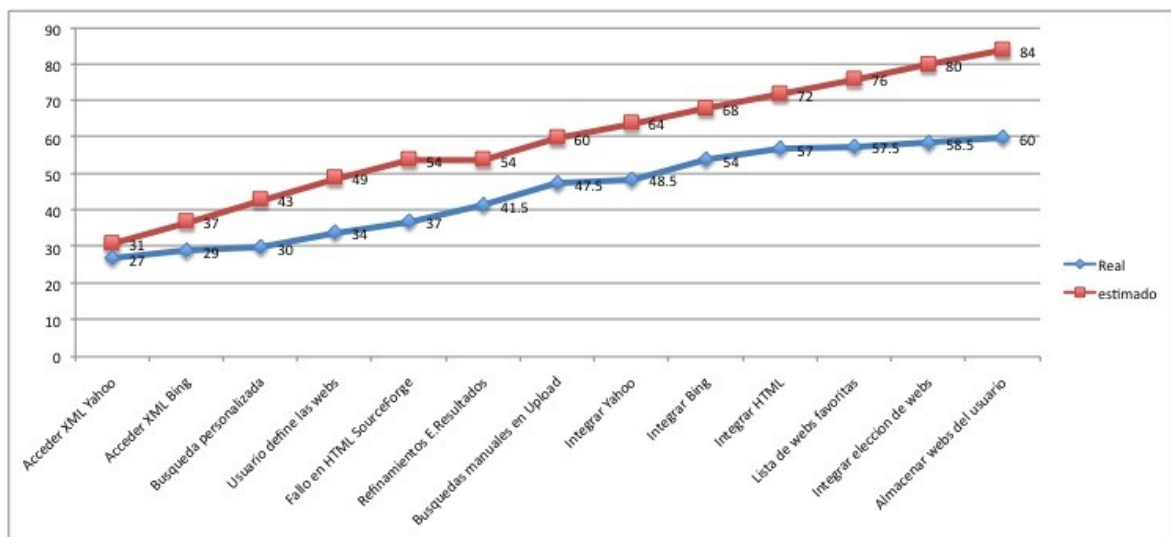
En la siguiente fotografía de la pizarra comprobamos que existían varias tareas que al inicio del sprint se encontraban ya en estado *in progress*. Estas eran las que quedaron pendientes en el anterior, como el refinamiento de la estructura de resultados. Dicha tarea se refiere a permitir al usuario añadir etiquetas y buscar manualmente referencias sin necesidad de volver a subir el proyecto, creando una nueva WebPart.





Fotografía 7.2 pizarra al inicio de Sprint parte dcha.

## 7.2 Seguimiento de las tareas



Gráfica 7.3 del sprint 3

En la gráfica se puede observar como se ha compensado el tiempo respecto al sprint anterior, porque en este caso, la línea de lo estimado queda bastante por encima del tiempo real. Vemos que hay una tarea de nombre *Fallo en HTML SourceForge*, esta no corresponde a ninguna historia de usuario concreta sino a un fallo que surgió y explicaré en las incidencias.

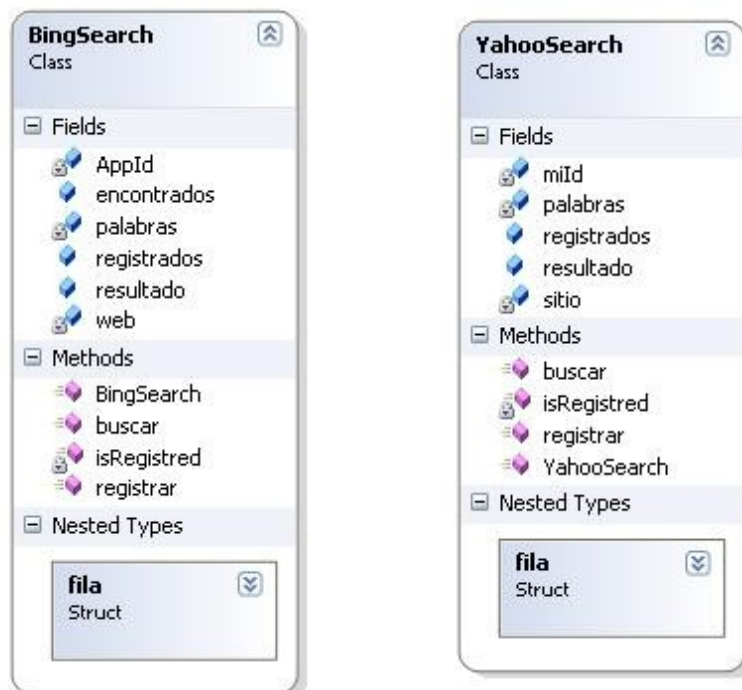
### 7.3 Diagramas de clase y listas creadas



Era necesario gestionar las posibles búsquedas que se dan en la aplicación: búsquedas Html para Codeplex y SourceForge, y búsquedas Xml para cualquier web desde Yahoo o Bing. Con este fin creé el gestor de búsquedas, que hace de intermediario entre la petición del usuario, la clase búsqueda desarrollada en el Sprint 1 y las dos siguientes clases.

Diagrama 7.4 Clase Gestorbúsqueda

Las dos siguientes clases 7.5 y 7.6, junto al gestor de búsquedas y a la clase de Html, las usaré en varias aplicaciones, como en la WebPart de añadir tags, o más adelante en el flujo de trabajo. Al igual que la clase de búsquedas Html estas también usan la estructura creada *fila*, así como la función registrar. Ambas poseen una variable constante que es la Id de la aplicación y otra de nombre *registrados* que es la información que devuelven al gestor de búsquedas.



Diagramas 7.5 y 7.6 Clases *BingSearch* y *YahooSearch*

El diagrama de clase de la WebPart de añadir etiquetas llama a las clases anteriores de búsquedas y, a través de los controles definidos, permite las consultas simultaneas sobre los tags de un proyecto. En un lado tenemos los campos y en otro los métodos:

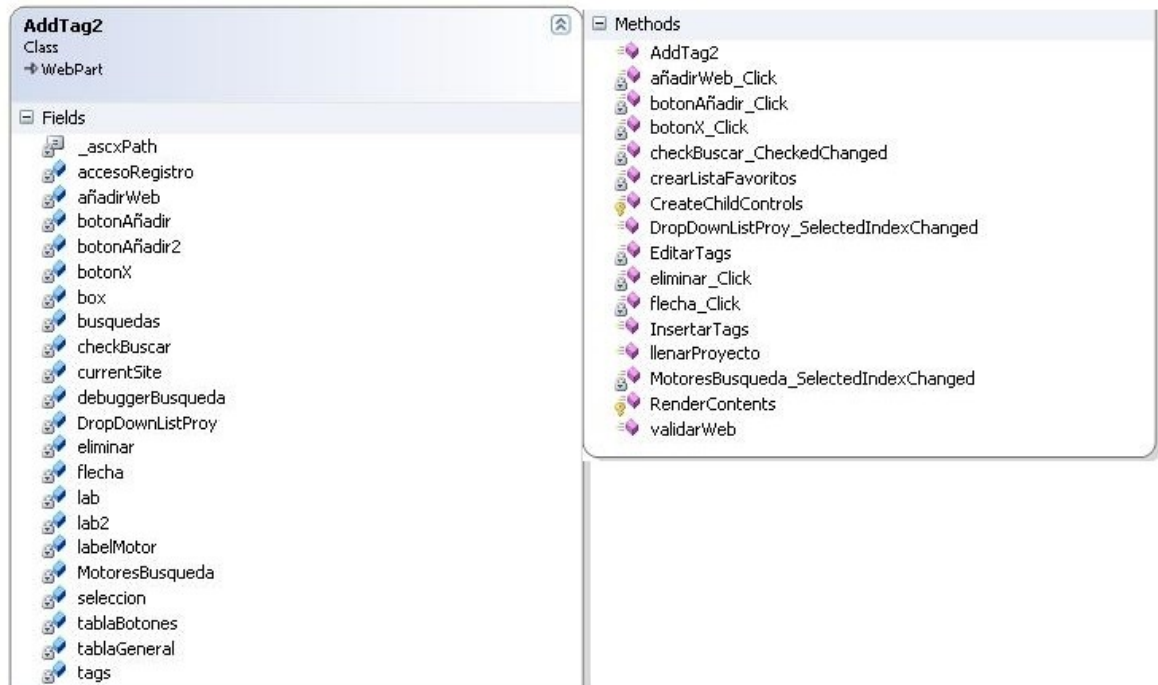


Figura 7.7 diagrama de clase AddTag

Creé una lista de páginas favoritas con el fin de almacenar las páginas indicadas por el usuario para las búsquedas y validadas por mi programa. Esta es muy sencilla y consta únicamente de dos columnas: título y motor de búsqueda.



Captura 7.8 Lista de páginas favoritas



## 7.4 Incidencias destacables

El fallo producido en el Html al que hago mención en la gráfica se refiere a que de un día a otro el buscador que implementé que leía el Html de SourceForge dejó de funcionar. Esto ocurrió debido a que cambiaron la web y por tanto el código fuente también. A pesar de que lo arreglé de forma bastante sencilla, el suceso me demostró que las búsquedas realizadas de este modo son muy inestables, además de estáticas como ya sabía, puesto que para cada página necesitas funciones diferentes para acceder a los datos. Por tanto las consultas Xml se impusieron como el mejor método de ambos para realizar búsquedas personalizadas.

Uno de los aspectos necesarios que fueron surgiendo durante el desarrollo fue la comprobación de que las páginas webs introducidas por el usuario fueran válidas. Para ello realicé dos comprobaciones. La primera dividir el string a partir del carácter punto, para analizar la estructura de la Url introducida. Si esta es correcta paso a comprobar que la Url devuelve algo usando la siguiente función:

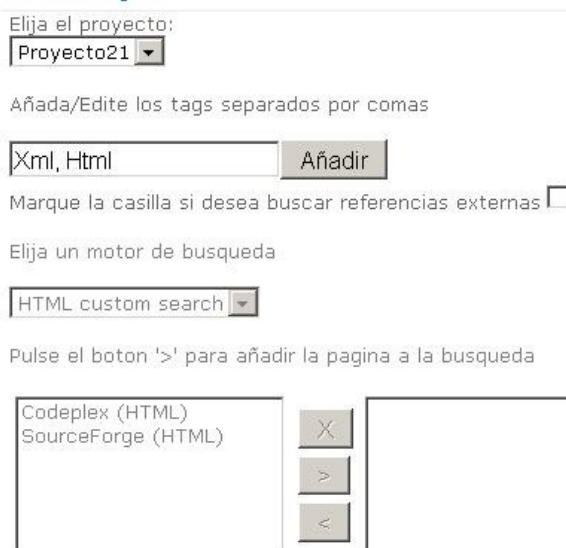
```
try
{
    if (tokens[0].Equals("http://www"))
        web = web.Replace("http://", "");
    IPHostEntry ipHost = Dns.GetHostEntry(web);
    return true;
}
catch (System.Net.Sockets.SocketException se)
{
    lab.Text = lab.Text + "La web no está disponible<br>";
    return false;
}
```

La manera de realizar las consultas Xml fue explicada en el capítulo de herramientas, aunque sólo expliqué para el buscador Bing, la Api de Yahoo es muy similar. Crear una aplicación en su web, obtener un Id y con él acceder a una web concreta que proporciona resultados de búsquedas en formato Xml. A la query se le puede añadir la web concreta en la que buscar.

## 7.5 Producto entregable

El elemento web *Añadir Tags* es uno de los más importantes puesto que engloba gran parte de la funcionalidad del proyecto: las consultas. Podemos ver en la captura que al inicio disponemos de un *DropList*, que indica que elijamos el proyecto, un texto editable donde debemos añadir o editar los tags, un botón y un *checkbox*. Más abajo hay una parte que no se encuentra activada, esta parte es la referente a las búsquedas y se activa cuando el *checkbox* se encuentra marcado.

### Añadir Tags



Captura 7.9 Añadir tags

Vamos a analizar más detalladamente cómo funciona. En la captura 7.10 de la siguiente página el usuario acaba de subir un proyecto. La WebPart de subida de proyectos contiene los mismos controles que la de añadir tags(fig. 7.9), estos se activan cuando un proyecto se ha cargado correctamente. En este caso hemos añadido las etiquetas *SharePoint*, *2010*, *C#* y hemos pulsado el botón de añadir, ignorando todo lo demás. Vemos en la parte de abajo de la captura de pantalla el mensaje: *Tags añadidos correctamente* redondeado.

En el caso de que hubiéramos modificado los tags actuales del proyecto, el mensaje preconfigurado sería *Tags modificados correctamente*. Esto ocurre,por ejemplo si lo hacemos desde la WebPart de añadir tags, o si subimos el mismo proyecto.

---

### Subir Proyecto

---

El proyecto se ha subido correctamente. Para visitarlo pulse [aquí](#).

[Volver a subir](#)

Añada los tags separados por comas

Marque la casilla si desea buscar referencias externas ☐

Elija un motor de búsqueda

Pulse el boton '>' para añadir la pagina a la busqueda

Codeplex (HTML)

SourceForge (HTML)

X

>

<

Tags añadidos correctamente

---

Captura 7.10 Añadiendo tags

Ahora bien, si marcamos el *checkbox*, como ya indica el Label pasaríamos a buscar referencias externas. Si el campo de etiquetas se encuentra en blanco no podremos marcar la casilla. Y nos lo indicaría en un mensaje de error.

Una vez escritas las etiquetas y marcada la casilla vemos en la captura de pantalla 7.11, que el botón que inicialmente se llamaba *Añadir*, ha pasado a localizarse en la parte de abajo, y ha cambiado su nombre por *Añadir y Buscar*. Esto quiere decir que cuando pulsemos el botón no sólo añadirá o modificará las etiquetas correspondientes, sino que además buscará referencias en las webs indicadas en el recuadro de la derecha.

En la figura encontramos dos flechas. La primera señala a la casilla o *checkbox*. La segunda flecha de la captura señala a la lista de motores de búsqueda que incluye: consultas Html, Yahoo y Bing. Correspondientes a las historias de usuario consultas Html y consultas Xml.

Subir Proyecto

El proyecto se ha subido correctamente. Para visitarlo pulse [aquí](#).

[Volver a subir](#)

Añada los tags separados por comas

SharePoint, 2010, C#

Marque la casilla si desea buscar referencias externas ☒

Elija un motor de búsqueda

Bing

Pulse el boton '>' para añadir la pagina a la búsqueda

elegir pagina..  
www.codeplex.com  
www.codeproject.com

X  
>  
<

SourceForge (HTML)  
Bing: www.codeproject.com

Añadir y Buscar

Captura 7.11 elección de consulta

El redondo de la captura 7.11 es el cuadro de control para la selección de consultas. En el cuadrado de la izquierda tenemos las webs almacenadas para el motor de búsquedas señalado por la flecha de arriba. Con las flechas podremos pasar de un lado a otro las páginas, las que estén en la derecha en el momento de pulsar el botón serán las webs sobre las que buscar. En los motores Bing y Yahoo se encuentra por defecto en la parte derecha la opción *elegir página..*, si llevamos esta a la derecha se abrirá un nuevo control para añadir webs y validarlas. El usuario podrá borrar estas páginas más tarde presionando *X*.

Elija un motor de búsqueda

Bing

Pulse el boton '>' para añadir la pagina a la búsqueda

elegir pagina..  
www.codeplex.com  
www.codeproject.com

X  
>  
<

Añadir y Buscar

Escriba la web que desea añadir

www.cesnavarra.net

Add web

Captura 7.12 Método de añadir webs

Las webs añadidas de esta forma por el usuario se almacenan para la siguiente búsqueda en la lista de *páginas favoritas*. Por último vemos el resultado de una búsqueda en varias páginas:

El proyecto se ha subido correctamente. Para visitarlo pulse [aquí](#).

[Volver a subir](#)

Añada los tags separados por comas

Sharepoint, XML

Marque la casilla si desea buscar referencias externas ☒

Elija un motor de búsqueda

Bing

Pulse el boton '>' para añadir la pagina a la busqueda

elegir pagina..	X	SourceForge (HTML)
www.codeplex.com	>	Yahoo: www.codeplex.com
www.codeproject.com	<	Bing: www.codeproject.com

[Añadir y Buscar](#)

Buscando en HTML custom los tags: Sharepoint, XML, con SourceForge (HTML)  
25 referencias encontradas  
23 referencias registradas  
Buscando en www.codeplex.com los tags: Sharepoint, XML, con Yahoo  
10 referencias encontradas  
6 referencias registradas  
Buscando en www.codeproject.com los tags: Sharepoint, XML, con Bing  
10 referencias encontradas  
9 referencias registradas

Tags añadidos correctamente

[38 referencias registradas](#)

Captura 7.13 resultado búsqueda

Si nos fijamos en la parte de abajo el programa nos indica todo lo que ha realizado. Ha buscado en las webs indicadas en la derecha: en SourceForge con el motor personalizado Html, en Codeplex con el motor Yahoo y en *www.codeproject.com* con el buscador Bing. En la primera ha encontrado 25 de los cuales ha registrado 23, puesto que los otros estaban registrados en la lista. En la segunda consulta registra 6 de 10 referencias encontradas, y 9 de la siguiente. Al final tenemos un hiper-vínculo que nos indica el número de referencias registradas, y al hacer click nos lleva al registro de búsquedas desde el cual podremos editarlas, aprobarlas, eliminarlas...

En el siguiente Sprint hice que tras la búsqueda pudiésemos seguir realizando consultas sobre el mismo proyecto sin necesidad de volver a subir el proyecto. Manteniendo los controles hasta que el usuario elige la opción volver a subir.



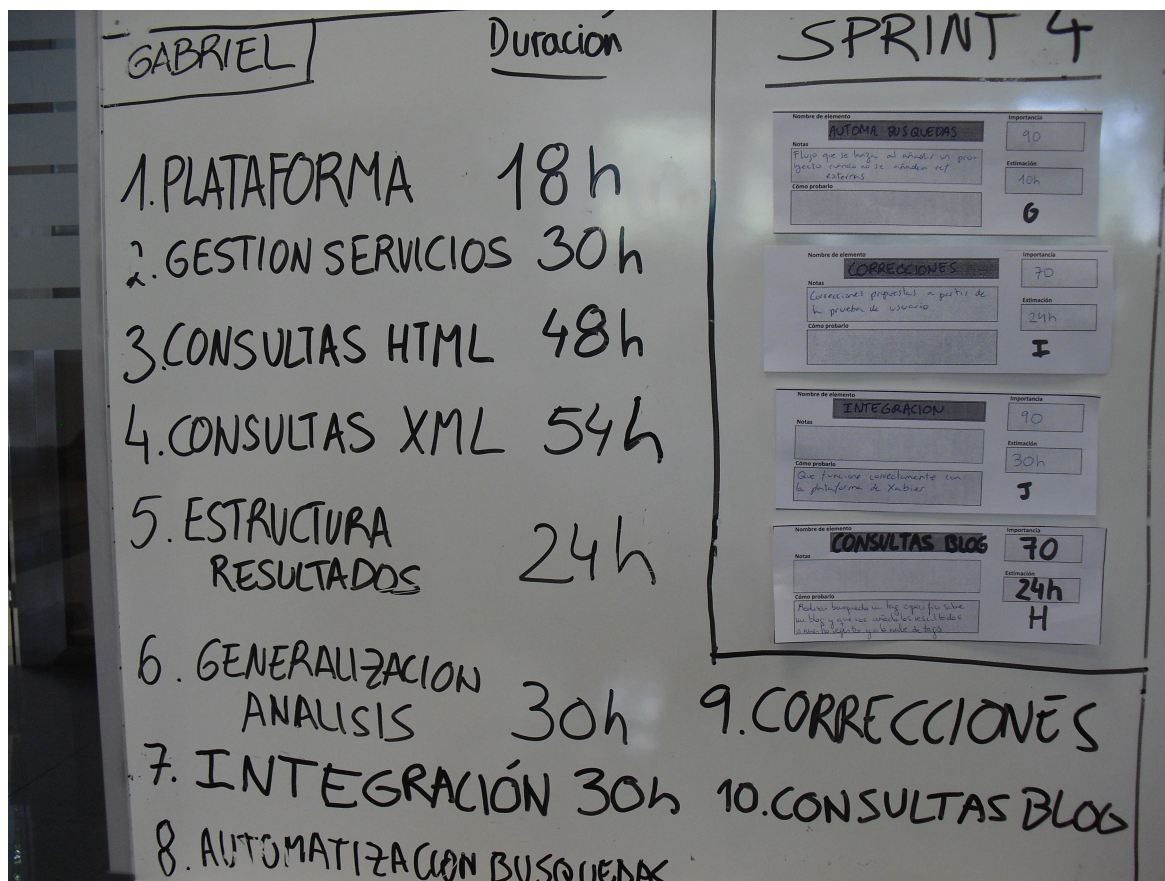
## 8. Desarrollo: Sprint 4

### 8.1 Selección de requisitos

A lo largo del desarrollo se producen inevitablemente cambios, el modelo de desarrollo utilizado trata de adaptar esos cambios. Al comienzo de este Sprint vimos necesario en las reuniones aumentar el product backlog para añadir dos historias de usuario nuevas.

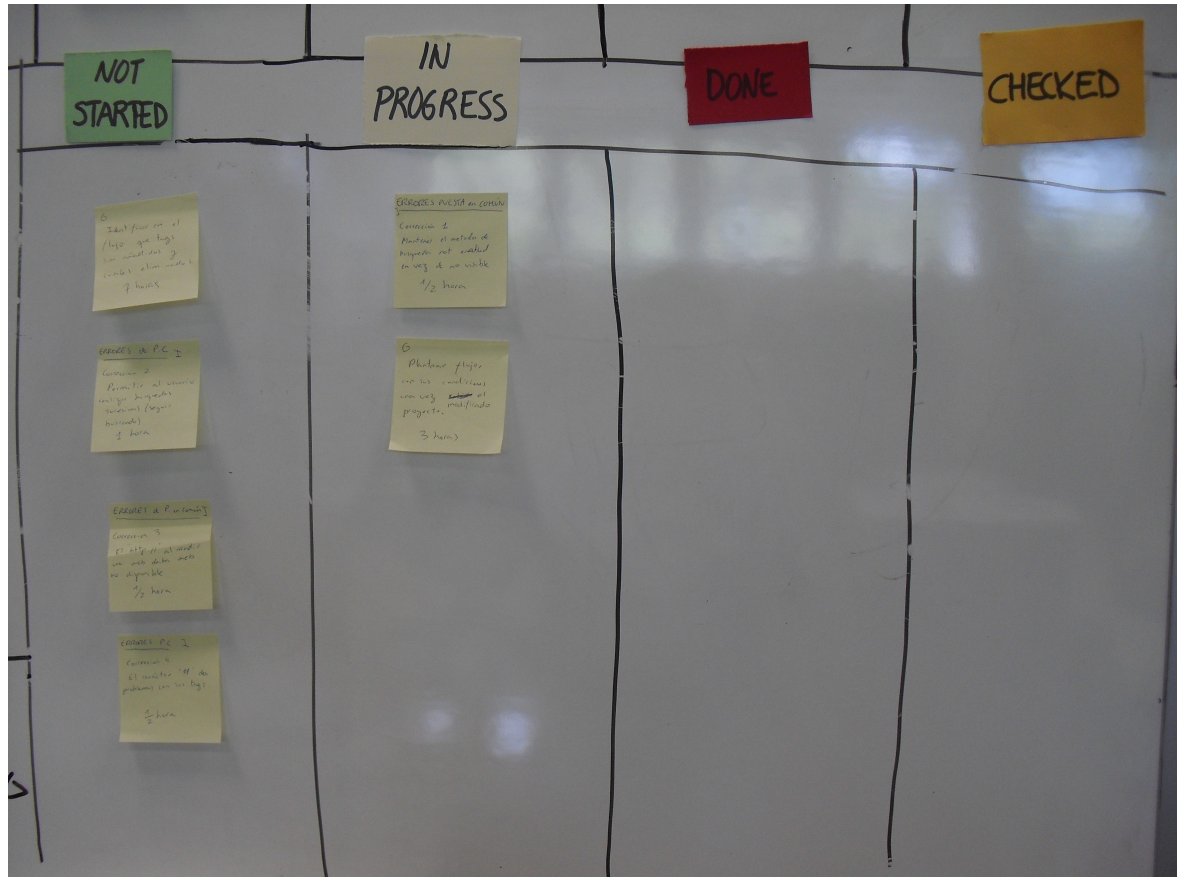
Título	Estimación	Importancia
Correcciones	24 h	70
Consultas Blog	24 h	70

La primera historia Correcciones se hizo tras hacer que otro compañero probara mi aplicación sin ningún tipo de indicación por mi parte. De este modo se observa desde un punto de vista objetivo en qué falla o qué se podría mejorar de una aplicación. Todo aquello que sugirió mi compañero como error lo incluí en una nueva historia de usuario, y por tanto se estimó y se le otorgó la importancia que consideramos en la reunión. Como este era el último Sprint, se incluyeron todas las historias de usuario restantes. En el caso de que alguna no se finalizara, se asegura un producto funcional y acabado, dispuesto para el cliente y que cumple con las historias de mayor importancia definidas por él; esto es maximización del valor. Las nuevas historias añadidas se colocan en la pizarra debajo de las anteriores( fotografía 8.1).



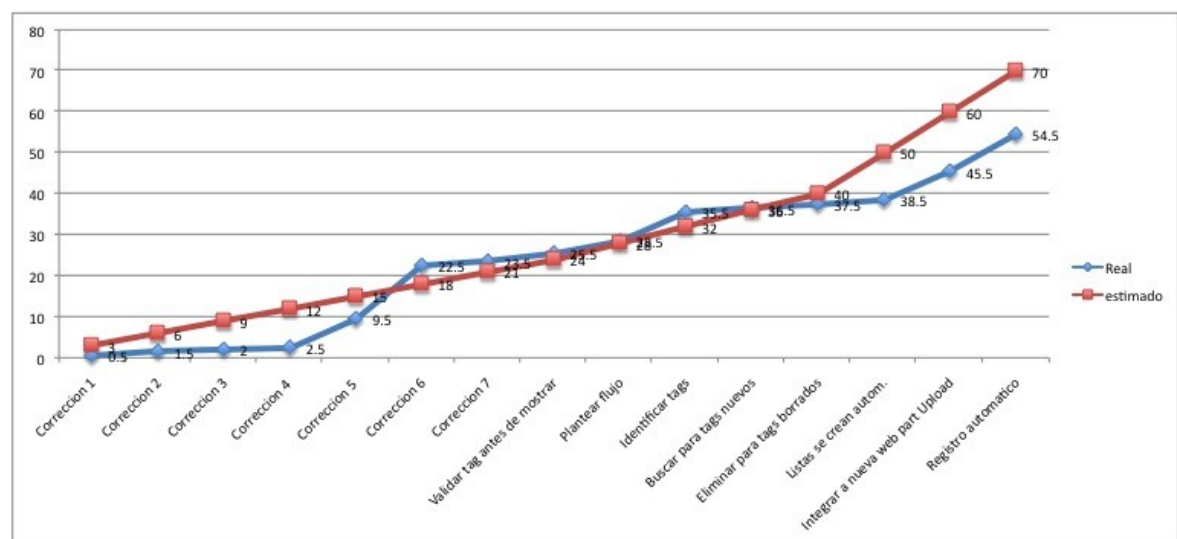
Fotografía 8.1 pizarra al inicio del Sprint 4

La historia de usuario de mayor importancia era la automatización de las búsquedas, que llevaba presente desde el inicio del proyecto, además las correcciones eran necesarias. Junto a la integración fueron las tres historias de usuario desarrolladas en este Sprint dejando fuera a las consultas de blog, que seguían pudiendo hacerse desde Xml pero no de manera personalizada como planteó el cliente.



Fotografía 8.2 pizarra parte dcha

## 8.2 Seguimiento de las tareas

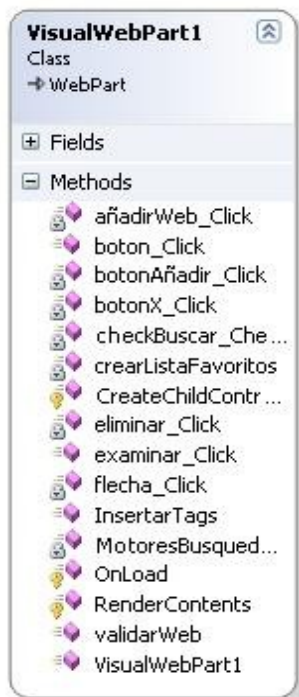


Gráfica 8.3 del Sprint 4



En este caso el tiempo estimado ha sido bastante preciso a lo largo del desarrollo, aunque cabe decir que en esta gráfica no se incluyen todas las historias de usuario, puesto que para la historia Consultas Blog no hubo tiempo. Las siete primeras tareas son correcciones, las cinco siguientes forman la creación e implantación del flujo de trabajo y las últimas obedecen a la historia de usuario Integración.

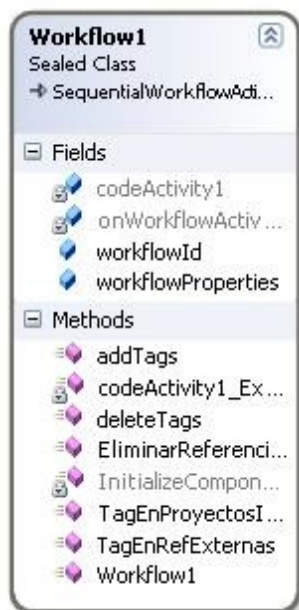
### 8.3 Diagramas de clase y listas creadas



Tras la integración de el módulo de inserción y búsqueda de referentes en la WebPart de subida de proyectos de mi compañero, la clase principal resultante es la de la izquierda.

Tiene métodos similares a la clase vista en el anterior capítulo de añadir tags. Pero además de estos posee alguno propio de la parte desarrollada por mi compañero.

Diagrama 8.4 clase Upload



Aquí podemos ver la clase principal del workflow o flujo de trabajo. Se encuentra conectada mi sitio de SharePoint, concretamente a la lista de proyectos, cuando un proyecto es modificado el flujo se activa.

Para cada tag añadido entrará en la función *addTags()* y para cada tag eliminado en *deleteTags()*, en el campo *workflowProperties* se guarda el valor del elemento de la lista que ha lanzado el workflow.

Diagrama 8.5 clase de FlujoTags

Para devolver la información proveniente del flujo creé una columna en la lista de

proyectos llamada *infoTags*, formada por varias líneas de texto. En esta se informa de las etiquetas que se han añadido y eliminado y muestra un link a los resultados para cada etiqueta nueva.

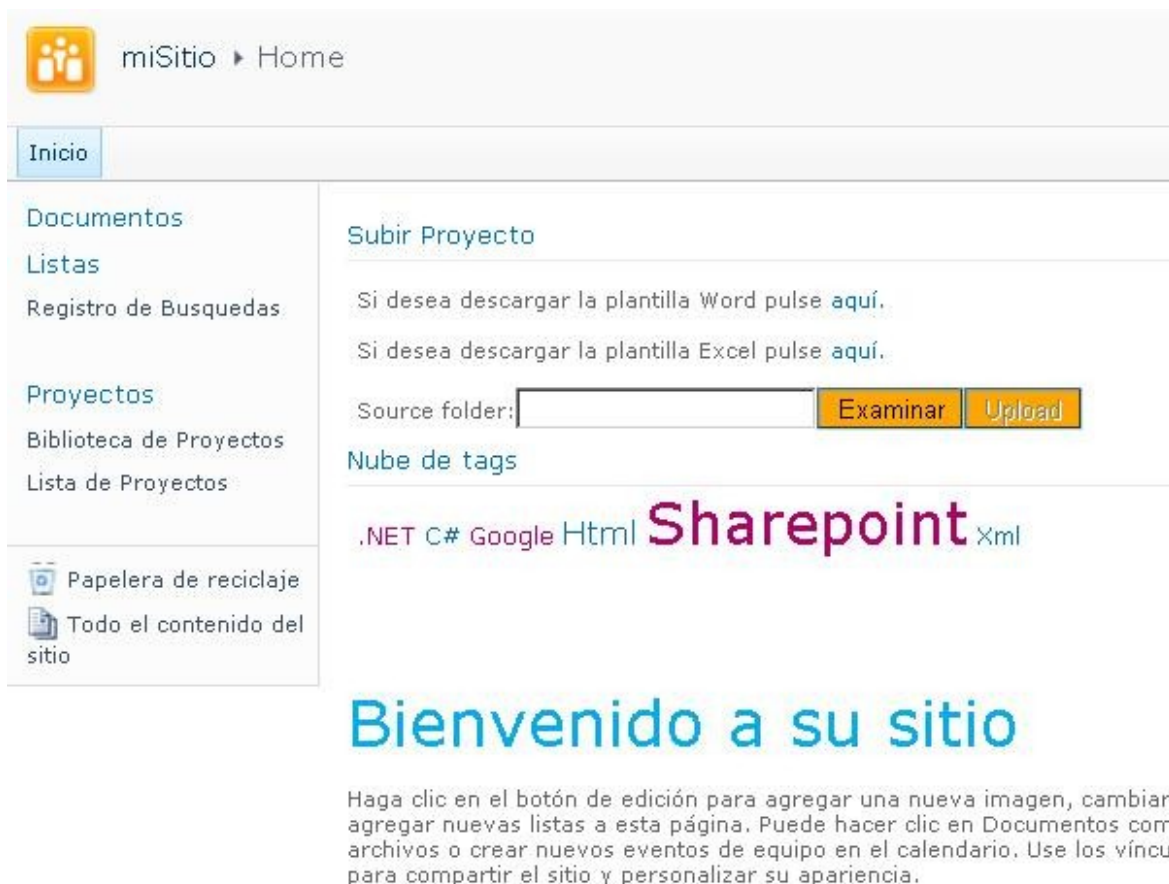
#### 8.4 Incidencias destacables

Al mismo tiempo que desarrollaba mi módulo, un compañero realizaba otro proyecto complementario al mio. Se trataba de la subida de los proyectos a la plataforma, validando su estructura antes de aceptarlos. Por tanto era lógico que integrase su WebPart de subida de proyectos con la funcionalidad de añadir etiquetas y buscar referencias. Esto formó parte de la historia de usuario Integración, donde además hice que mis listas se crearan automáticamente con la instalación de las WebParts.

Este mismo compañero fue con quien hice las pruebas de puesta en común. Desde nuestro punto de vista objetivo, cada uno de nosotros probó la aplicación del otro en busca de posibles mejoras. A esto corresponde las siete primeras tareas de la gráfica, donde traté errores de casos aislados como caracteres que no funcionaban bien en la nube. O errores de eficiencia, como validar si existe una etiqueta antes de crear una vista vacía. De este modo si el usuario ha buscado para una etiqueta que no está en un proyecto devuelvo que no se ha encontrado ninguna referencia. Y también errores de controles poco intuitivos, esta visión quien mejor la puede aportar es el usuario, en este caso mi compañero actuaba como tal.

El flujo de trabajo inicialmente pensado iba a ser muy sencillo, cuando se introdujera una etiqueta a un proyecto y el usuario decida no buscar manualmente para esta, el flujo lo haría usando las consultas implementadas. Al final el flujo quedo algo más complejo, puesto que si se elimina una etiqueta en todos los proyectos, también se eliminan todos sus referentes externos asociados. Esto se explica mejor en el capítulo de herramientas, y más abajo en el producto entregable se visualiza a través de capturas de pantalla.

## 8.5 Producto entregable



Captura 8.6 Pantalla inicio

La captura anterior es la página principal de la Plataforma tras la integración con la WebPart de subida de proyectos. A la izquierda tenemos las listas que usa el usuario, a la derecha la nube de tags.

Ahora mostraré como actúa el flujo sobre la lista de proyectos, para empezar podemos ver la vista de dicha lista en la captura 8.7:



Captura 8.7 Lista de Proyectos

Tanto si se modifican la columna *Tags* desde la propia lista o desde las WebPart de modificación de tags (añadir Tags o subir proyecto) el flujo se lanzará.

En el primer ejemplo elimino la etiqueta *web* del proyecto cuya *ID Proyecto* es 2. Esta etiqueta no tiene ninguna referencia externa asociada por lo tanto el flujo la borrará sin más. Podemos ver el resultado en la columna *Info de Tags* de la siguiente captura.

Tags	Info de Tags	FlujoTags
Sharepoint, .NET		
Xml, Html	Tags eliminados: web	Finalizado
SharePoint, C#		

Captura 8.8 tag eliminado

Ahora el proyecto contiene los tags *Xml* , *Html*. Si añadimos uno nuevo y no buscamos referencias externas manualmente, el flujo se encargará de hacerlo. En el caso de la figura 8.9 añadimos una nueva etiqueta *language*:

Tags	Info de Tags	FlujoTags
Sharepoint, .NET		
Xml, Html, language	Tags añadidos: Buscando referentes para language 0 resultados de búsqueda en www.msdn.com con Yahoo 10 resultados de búsqueda en www.codeplex.com con Bing 10 resultados de búsqueda en www.sourceforge.net con Yahoo language	Finalizado
SharePoint, C#		

Captura 8.9 tag añadido

El flujo ha buscado en las tres primeras páginas de la lista de páginas favoritas del usuario. Esto lo configuré así aunque se puede cambiar para las webs que deseemos. Devuelve en último lugar un link ,con el nombre de la etiqueta, a la página de resultados para dicha etiqueta.

Ahora si decido borrar la etiqueta *language* que sabemos que posee referencias debería eliminar todas ellas, y además incluyo la etiqueta *SharePoint* común a otros proyectos:

Tags	Info de Tags	FlujoTags
Sharepoint, .NET		
Xml, Html, SharePoint	Tags eliminados: language (Eliminados referentes externos) Tags añadidos: SharePoint	Finalizado
SharePoint, C#		

Captura 8.10 tag añadido y tag eliminado

El flujo nos indica que ha eliminado los referentes externos para *language*, y sin embargo no realiza ninguna búsqueda para la etiqueta *SharePoint*, esto es debido a que ya posee referencias externas.

## 9. Conclusiones y líneas futuras

### 9.1 Conclusiones

El uso de Scrum como metodología ha supuesto la constante entrega de productos funcionales. Esto en cualquier proyecto informático reduce el riesgo del cliente, puede estar seguro que cuando finalice el tiempo recibirá un producto funcional que le aporta valor. Además la implicación del cliente en el desarrollo, provoca que el producto sea flexible durante su desarrollo a las necesidades y cambios que van surgiendo.

El módulo de localización e inserción de referentes externos, que he desarrollado, es un producto terminado y funcional, cumple con los requisitos que el cliente ha ido marcando como prioritarios. Aunque para la última historia de usuario no dio tiempo, las demás historias se desarrollaron teniendo en cuenta en todo momento el concepto de calidad. Este módulo tiene por objetivos ayudar al desarrollador en su ardua tarea. Para ello se vale de la puesta en común, de la búsqueda y clasificación de proyectos similares a través de etiquetas.

El etiquetado es el presente de la red, una red que evoluciona constantemente hacia la web semántica. Mi proyecto es un paso hacia ésta. El esfuerzo del usuario es necesario, a la hora de aprobar los referentes encontrados en la red. Esto hace de la plataforma un nexo entre el desarrollador y la propia web. Los flujos de trabajo en SharePoint son una herramienta muy útil en cualquier gestión. La automatización del proceso estuvo presente como idea desde el inicio de las tareas. Las búsquedas y eliminaciones desarrolladas automáticamente ayudan al dinamismo de la plataforma de conocimiento.

Una conclusión importante que saco con mi proyecto es la inconveniencia de las búsquedas realizadas puramente en Html. Las que he desarrollado han ido fallando cada vez que el código cambiaba. Su inestabilidad y su falta de dinamismo, puesto que son distintas para cada página, hacen que las consultas Xml sean muy superiores.

Mi valoración de la plataforma SharePoint es positiva. La variedad y flexibilidad de las soluciones que podemos crear es enorme. La integración de ésta con Visual Studio nos permite crear aplicaciones de forma sencilla. La última versión 2010 propone grandes mejoras respecto a la anterior. Sin embargo el hecho de haber trabajado en una versión beta, supuso algún inconveniente a la hora de desarrollar software para esta.

Para desarrollar este proyecto he recibido una ayuda fundamental tanto del tutor como de todos los integrantes del proceso; scrum master y compañeros. Las reuniones semanales y los controles diarios me brindaban siempre un punto de vista diferente, además de soluciones a los obstáculos que iban surgiendo durante el desarrollo.

En cuanto a las metodologías ágiles, el balance que saco es muy positivo. El desarrollo de mi proyecto en el CEIN me ha dado la posibilidad de asistir a charlas sobre este tema. Scrum como proceso, nos proporciona unas herramientas excelentes, centradas en el trabajo en equipo. Sin embargo si no se interiorizan los principios clave, estas herramientas pueden ser inservibles.



## 9.2 Líneas Futuras

Un concepto, que he citado previamente, y que mejoraría notablemente la aplicación es la semántica[8]. Una web semántica es aquella que además de procesar la información la interpreta. Para que ésto ocurra, es necesario que la información de las páginas web se codifique mediante ontologías. Las ontologías representarán el conocimiento de Internet, definiendo formalmente los conceptos de los diferentes dominios y sus relaciones, con capacidad para realizar deducciones con este conocimiento.

Al módulo desarrollado se le podría añadir dicha posibilidad de interpretar la información alojada, con el fin de generar un etiquetado en función del significado. De este modo la intervención del usuario se reduciría a subir sus proyectos y recibir la mejor información personalizada del propio software.

La última historia de usuario, consultas de Blogs, no entró en el desarrollo por falta de tiempo. Aunque es verdad que desde las consultas Xml podemos consultar en cualquier sitio incluidos blogs, la idea era una búsqueda generalizada sobre los principales blogs relacionados con el desarrollo de software y la informática. Estas consultas podrían desarrollarse como las creadas para Codeplex y SourceForge, desde Html. Sin embargo la experiencia pasada lo desaconseja por su inestabilidad.

Además un asunto pendiente de este proyecto es la integración con la plataforma de conocimiento anterior. Ésta fue desarrollada en un servidor bajo SharePoint Server 2007. No pude realizar la restauración al nuevo sistema operativo, debido a que mi maquina no cumplía los requisitos mínimos de espacio para la base de datos. Por tanto, desarrollé el proyecto sobre un nuevo sitio, dejando pendiente para el futuro añadir la aplicación al sitio.

## 10. Bibliografía

### Referencias webs

[1]

Módulo de automatización de búsquedas con Scrum

CES Digital Blog

<http://www.cesnavarra.net/cesdigitalblog/Lists/Entradas%20de%20blog/Post.aspx?ID=57>

[2]

Vmware

Wmware- Wikipedia, la enciclopedia libre

<http://es.wikipedia.org/wiki/VMware>

[3]

Arquitectura básica de SharePoint

SkunkWorks SharePoint Portal

[http://www.gavd.net/servers/sharepointv4/spsv4\\_item.aspx](http://www.gavd.net/servers/sharepointv4/spsv4_item.aspx)

[4]

Bing Api, Versión 2

Msdn

<http://msdn.microsoft.com/en-us/library/dd251056.aspx>

[5]

Introducción a Lean y Kanban- Slideshare

Proyectails

<http://www.slideshare.net/proyectalis/100426-lean-thinking-10print>

[6]

Qué es Scrum

Proyectos Ágiles

<http://www.proyectosagiles.org/que-es-scrum>

[7]

Community Kit for SharePoint

Codeplex

<http://cks.codeplex.com/>

[8]

Web semantica- PDF

I Jornadas Internacionales sobre Ingeniería Web

[www.informandote.com/jornadasIngWEB/articulos/jiw02.pdf](http://www.informandote.com/jornadasIngWEB/articulos/jiw02.pdf)

## **Libros**

Sharp, J.; Jagger, J., (2002), *Aprenda ya Microsoft Visual C# .NET*, McGrawHill, Madrid.

Holliday, J.; Alexander, J.; Julian, J.; Robillard, E.; Schwartz, B; Ranlett, M; Dan Attis, J.; Buenz, A.; Rizzo, T., (2007), *Professional SharePoint 2007 Development*, Wiley Publishing, Inc. , Indiana.

# Proyecto fin de carrera

MÓDULO DE LOCALIZACIÓN E INSERCIÓN AUTOMÁTICA  
DE REFERENCIAS EXTERNAS ASOCIADAS AL CÓDIGO  
PUBLICADO EN WEBS DE BUENAS PRÁCTICAS DE  
DESARROLLO

# Introducción



- Plataforma de Conocimiento
- Etiquetado
- Búsquedas en webs de buenas prácticas
- Automatización



# Introducción

Microsoft®  
**SharePoint**2010

- Visual Studio 2010
- Scrum

# Scrum

- Metodologías Ágiles

*We are uncovering better ways of developing software by doing it and helping others do it.*

*Through this work we have come to value:*

***Individuals and interactions over processes and tools***

***Working software over comprehensive documentation***

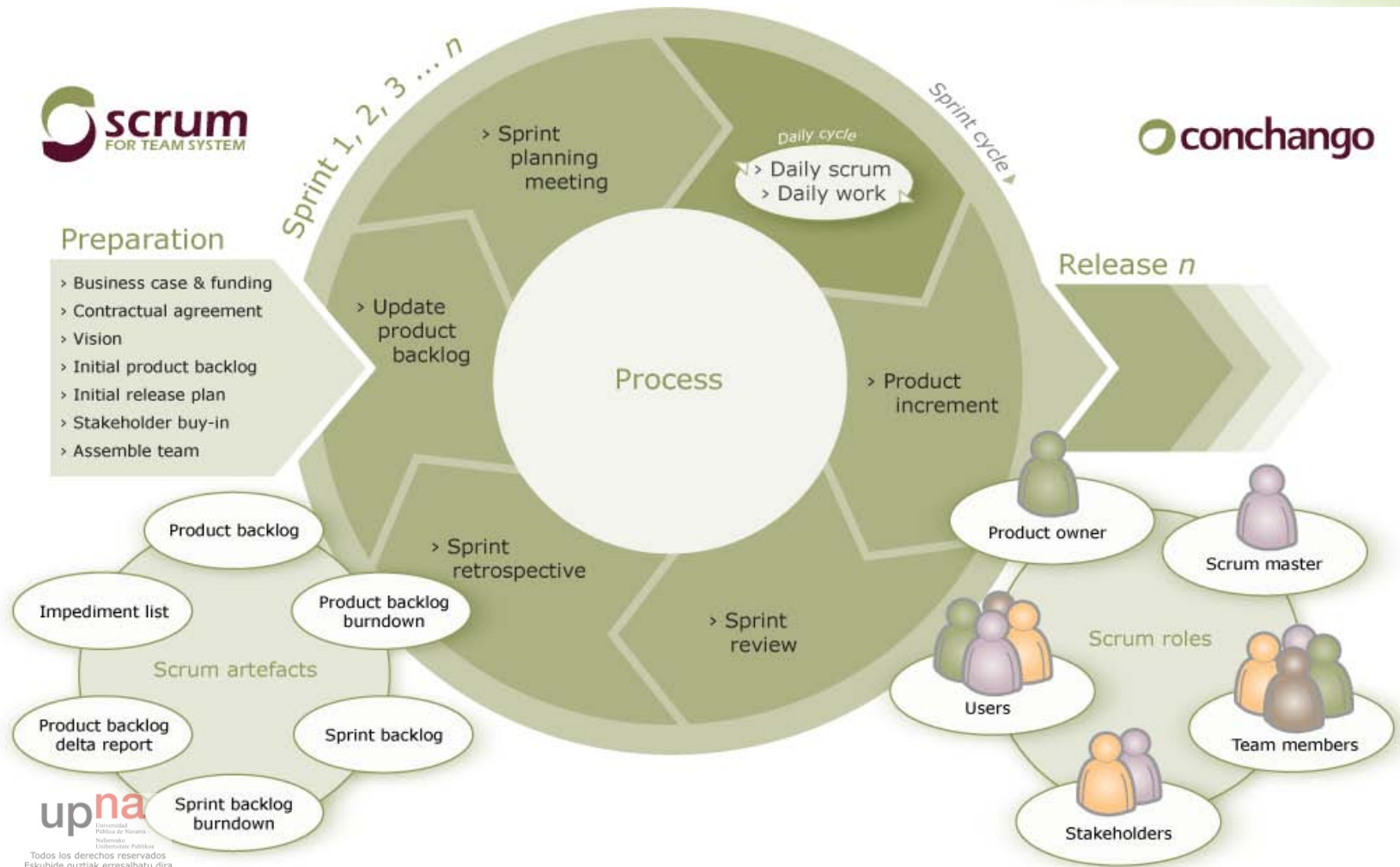
***Customer collaboration over contract negotiation***

***Responding to change over following a plan***

*That is, while there is value in the items on the right, we value the items on the left more.*

# Scrum

- Iteraciones: Sprints



# Scrum

- Product Backlog
- Roles asignados



# Desarrollo: esquema

- Selección de Requisitos
- Gráfica de seguimiento
- Producto entregable



# Sprint 1

## Selección de Requisitos

PLATAFORMA  
GESTIÓN SERVICIOS  
CONSULTAS HTML

SPRINT 1

Nombre de elemento	Importancia
PLATAFORMA/SET UP	50
Notas	Información
Cómo probarlo	18h
AVANZA Y PARA TODO CORRECTAMENTE	★

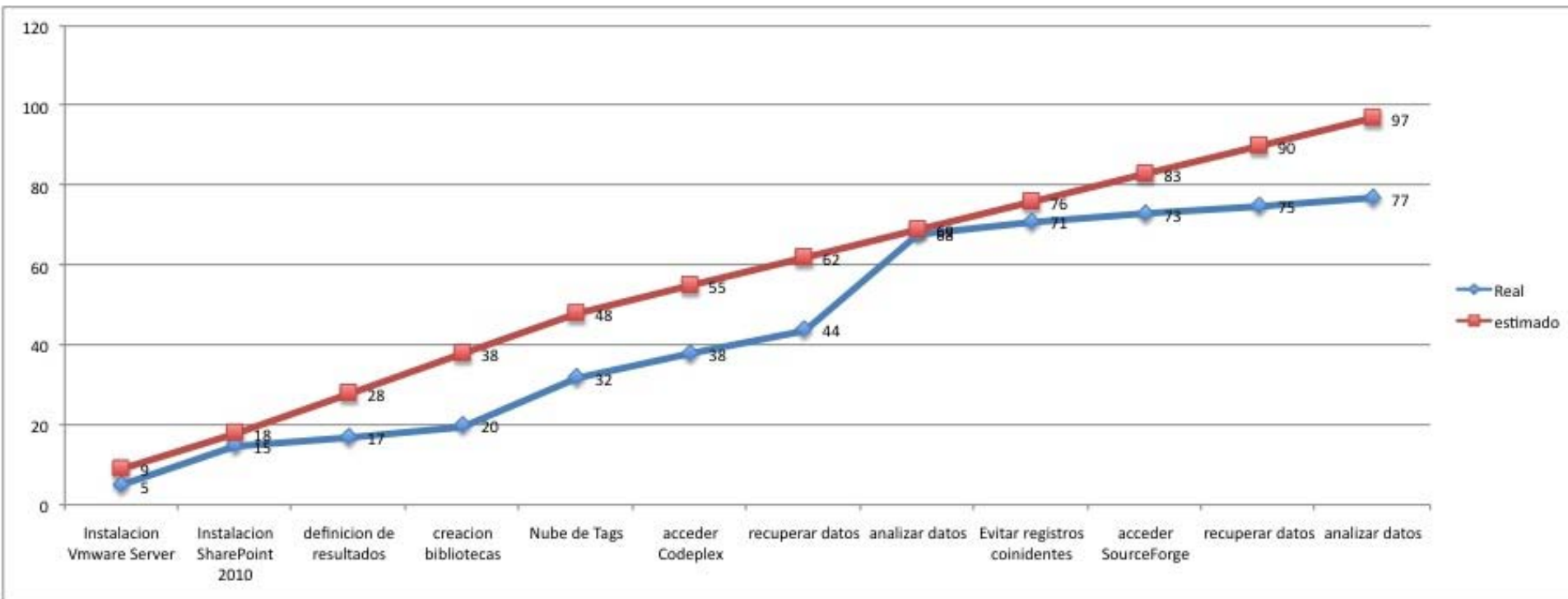
Nombre de elemento	Importancia
GESTIÓN SERVICIOS	70
Notas	Información
Cómo probarlo	30h
QUE SERVICIOS USAR A OTRAS QUE CONTIENE LOS OPACOS ESTABLECER DINAMICA API SE ALISTE INSTALAR Y DESARROLAR UN SERVICIO DE CONSULTA NUEVA Y PARA DE DATA DATOS	B

Nombre de elemento	Importancia
CONSULTAS HTML	100
Notas	Información
Cómo probarlo	48h
UN TERMINO GENERA RESULTADO Y UN TERMINO OTRO NO GENERA NADA	C

# Sprint 1

## Seguimiento de las tareas



# Sprint 1

miSitio > Home

Inicio

Documentos

Listas

Registro de Búsquedas

Proyectos

Biblioteca de Proyectos

Lista de Proyectos

Papelera de reciclaje

Todo el contenido del sitio

Nube de tags

.NET 2010 Html **Sharepoint** **SharePoint** 1 regs

Bienvenido a su sitio

Haga clic en el botón de edición para agregar una nueva imagen, cambiar el texto de bienvenida o agregar nuevas listas a esta página. R

archivos o crear nuevos eventos de e

inicio para compartir el sitio y persona

Acciones del sitio Examinar Página

miSitio > resultado

Inicio

Documentos

Listas

Registro de Búsquedas

Proyectos

Biblioteca de Proyectos

Lista de Proyectos

Papelera de reciclaje

Todo el contenido del sitio

Referencias internas

Tagged items for Sharepoint:

PFC2  
Created on 25/06/2010 Saved 1 times.  
Tags: SharePoint, 2010

PFC  
Created on 22/06/2010 Saved 1 times.  
Tags: Sharepoint, .NET

Tags relacionados

.NET 2010

Tags externos

.NET Html Sharepoint Xml

# Sprint 1

## Registro de búsquedas

miSitio ▶ RegistroBusqueda : Vista del Registro ▼ Registro de las búsquedas de proyectos externos						
Inicio	Search this site... 🔍 ?					
Documentos	<input type="checkbox"/>	Título	descripcion	URL	Aprobacion	posibles tags
Listas		XAware Data Integration Project	Create services to integrate applications & move data of any type. Build data &quot;views&quot; across DBMS, SOAP, HTTP/REST, Salesforce, SAP, Microsoft, SharePoint, Text, LDAP, FTP sources to read, write & transfer data. Eclipse designer & runtime engine.	<a href="http://www.sourceforge.net/projects/xaware/">www.sourceforge.net/projects/xaware/</a>	Pendiente	Sharepoint, .NET
Registro de Búsquedas		UbuntuZilla: Mozilla Software Installer	An APT repository hosting the Mozilla builds of the latest official releases of Firefox, Thunderbird, and Seamonkey. Do not manually download the files - read the website, <a href="http://ubuntuzilla.sourceforge.net">http://ubuntuzilla.sourceforge.net</a> for instructions on using the repository.	<a href="http://www.sourceforge.net/projects/ubuntuzilla/">www.sourceforge.net/projects/ubuntuzilla/</a>	Pendiente	Sharepoint, .NET
Proyectos		TripleA Maps	A project by Veqryn for creating, centralizing, and hosting all open-source TripleA maps, as well as other developer resources such as artwork and map makers. If you need the actual TripleA program, go here: <a href="https://sourceforge.net/projects/triplea/">https://sourceforge.net/projects/triplea/</a>	<a href="http://www.sourceforge.net/projects/tripleamaps/">www.sourceforge.net/projects/tripleamaps/</a>	Pendiente	Sharepoint, .NET
Biblioteca de Proyectos		The Tao Framework	The Tao Framework for .NET is a collection of bindings to facilitate cross-platform game-related development utilizing the .NET platform.	<a href="http://www.sourceforge.net/projects/taoframework/">www.sourceforge.net/projects/taoframework/</a>	Pendiente	Sharepoint, .NET
Lista de Proyectos		StarDict	StarDict is a Cross-Platform and international dictionary written in Gtk2. It has powerful features such as &quot;Glob-style pattern matching&quot;, &quot;Scan selection word&quot;, &quot;Fuzzy search&quot;, &quot;Net dict&quot;, &quot;Full text translation&quot; etc.	<a href="http://www.sourceforge.net/projects/stardict/">www.sourceforge.net/projects/stardict/</a>	Pendiente	Sharepoint, .NET
Papelera de reciclaje		ADO.NET 2.0 Provider for SQLite	ADO.NET 2.0/3.5 Provider for SQLite. Supports Full and Compact Framework, Entity Framework, complete with full Visual Studio 2005/2008 design-time support.	<a href="http://www.sourceforge.net/projects/sqlite-dotnet2/">www.sourceforge.net/projects/sqlite-dotnet2/</a>	Pendiente	Sharepoint, .NET
Todo el contenido del sitio						

# Sprint 2

## Selección de Requisitos

### ESTRUCTURA DE RESULTADOS CONSULTAS XML

SPRINT 2

Nombre de elemento	Importancia
CONSULTAS XML	100
Notas	Estimación
	54h
Cómo probarlo	D
Un término genera resultados, uno vacío no genera nada.	

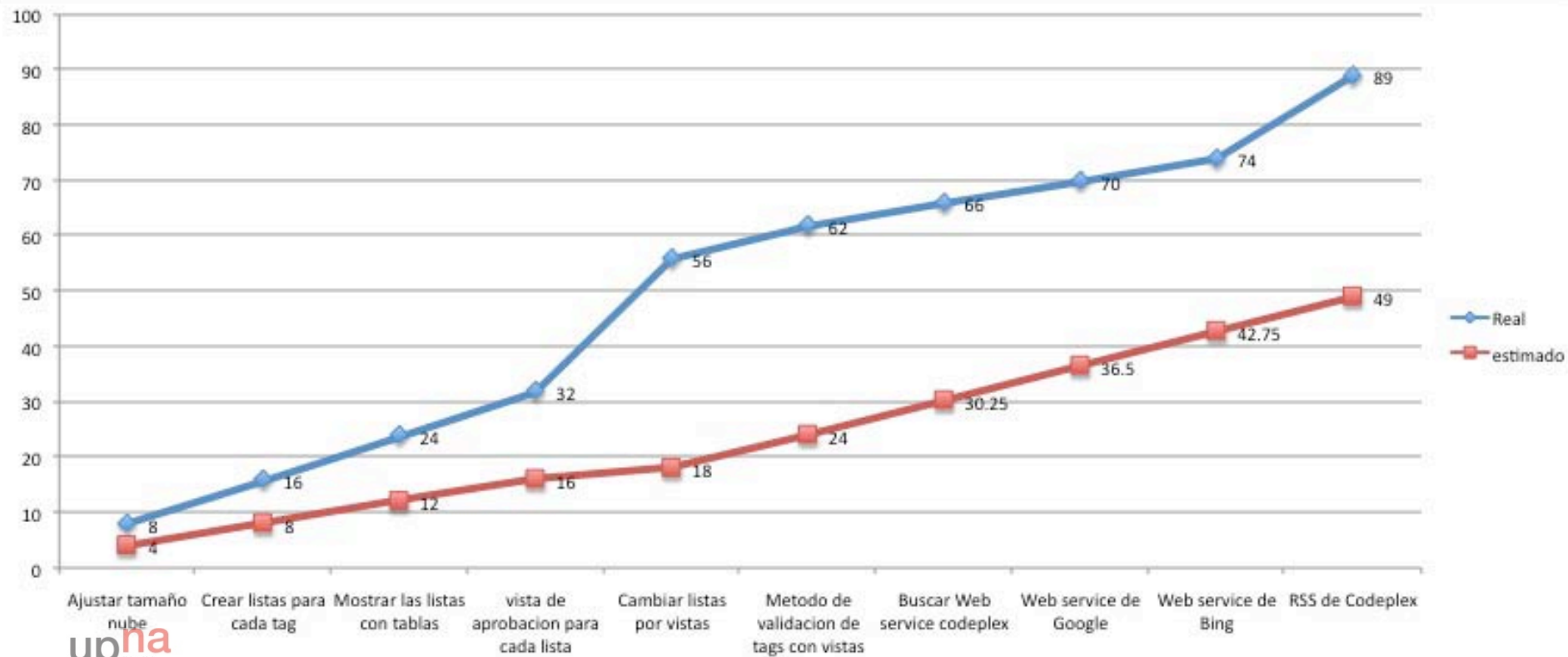
  

Nombre de elemento	Importancia
ESTRUCTURA DE RESULTADOS	80
Notas	Estimación
	24h
Cómo probarlo	E
Verificar que obtenemos con cada búsqueda los campos considerados como obligatorios.	




# Sprint 2

## Seguimiento de las tareas



# Sprint 2

## Consultas XML sin finalizar Navegación de referentes:

 miSitio ▶ resultado

[Inicio](#)

Documentos

Listas

Registro de Búsquedas

Proyectos

Biblioteca de Proyectos

Lista de Proyectos

Papelera de reciclaje

Todo el contenido del sitio

Referencias internas

Tagged items for **Sharepoint**:

PFC2

Created on 25/06/2010 Saved 1 times.

Tags: SharePoint, 2010

PFC

Created on 22/06/2010 Saved 1 times.

Tags: Sharepoint, .NET

Enlaces Externos

Referencias encontradas para: **Sharepoint**

Mostrando vista de referentes aprobados


Tags relacionados

**.NET**2010

Tags externos

**.NET**HtmlSharepointXml

Título	URL	descripcion
iText.NET	<a href="http://www.sourceforge.net/projects/itextdotnet/">www.sourceforge.net/projects/itextdotnet/</a>	iText.NET is iText porting on .NET Framework by J#. This library allows you to generate PDF, (X)HTML, XML, RTF files on Microsoft.NET Framework including ASP.NET.
HP Linux Imaging and Printing	<a href="http://www.sourceforge.net/projects/hplip/">www.sourceforge.net/projects/hplip/</a>	The Hewlett-Packard Linux Imaging and Printing Project provides a unified single and multi-function connectivity solution for Linux. For support and help, please visit <a href="https://launchpad.net/hplip">https://launchpad.net/hplip</a>
Your Development Resource - CodeProject	<a href="http://www.codeproject.com/">http://www.codeproject.com/</a>	Integrating Sharepoint 2010 and SQL Reporting Services 2008 in 6 easy steps [Technical Blog] Raymund Macaala [unedited] 17 Jun 2010 (new) Observer in .NET 4.0 with IObserver(T ...
Extended Interface for Toolbars - CodeProject	<a href="http://www.codeproject.com/KB/menus/extendedtoolbar.aspx">http://www.codeproject.com/KB/menus/extendedtoolbar.aspx</a>	Select the tab .NET Framework Components; Click Browse... Click on ExtendedInterface.DLL; Click OK. The ToolBar should have two new components: StatusMessage and ToolBarFunction.

 [Aprobar/Editar referencias](#)

Todos los derechos reservados  
Eskubide guztiak erresaltatu dira

Titulo	URL	descripcion
iText.NET	<a href="http://www.sourceforge.net/projects/itextdotnet/">www.sourceforge.net/projects/itextdotnet/</a>	iText.NET is iText porting on .NET Framework by J#. This library allows you to generate PDF, (X)HTML, XML, RTF files on Microsoft.NET Framework including ASP.NET.
HP Linux Imaging and Printing	<a href="http://www.sourceforge.net/projects/hplip/">www.sourceforge.net/projects/hplip/</a>	The Hewlett-Packard Linux Imaging and Printing Project provides a unified single and multi-function connectivity solution for Linux. For support and help, please visit <a href="https://launchpad.net/hplip">https://launchpad.net/hplip</a>
Your Development Resource - CodeProject	<a href="http://www.codeproject.com/">http://www.codeproject.com/</a>	Integrating Sharepoint 2010 and SQL Reporting Services 2008 in 6 easy steps [Technical Blog] Raymund Macaalay [unedited] 17 Jun 2010 (new) Observer in .NET 4.0 with IObserver(T ...
Extended Interface for Toolbars - CodeProject	<a href="http://www.codeproject.com/KB/menus/extendedtoolbar.aspx">http://www.codeproject.com/KB/menus/extendedtoolbar.aspx</a>	Select the tab .NET Framework Components; Click B rowse... Click on ExtendedInterface.DLL; Click OK. The Toolbox should have two new components: StatusMessage and ToolbarFunction.

Aprobar/Editar referencias

Seleccione los que desee aprobar y pulse el boton

Título	URL	description	posibles tags	Aprobación
UbuntuZilla: Mozilla Software Installer	<a href="http://www.sourceforge.net/projects/ubuntuzilla/">www.sourceforge.net/projects/ubuntuzilla/</a>	An APT repository hosting the Mozilla builds of the latest official releases of Firefox, Thunderbird, and Seamonkey. Do not manually download the files - read the website, <a href="http://ubuntuzilla.sourceforge.net">http://ubuntuzilla.sourceforge.net</a> for instructions on using the repository.	Sharepoint, .NET	Pendiente
net-snmp	<a href="http://www.sourceforge.net/projects/net-snmp/">www.sourceforge.net/projects/net-snmp/</a>	net-snmp provides tools and libraries relating to the Simple Network	Sharepoint, .NET	Pendiente
emule Xtreme Mod	<a href="http://www.sourceforge.net/projects/emulextreme/">www.sourceforge.net/projects/emulextreme/</a>	emule client, based on official emule version (see <a href="http://www.emule-project.net">www.emule-project.net</a> )	Sharepoint, .NET	Pendiente
StarDict	<a href="http://www.sourceforge.net/projects/stardict/">www.sourceforge.net/projects/stardict/</a>	StarDict is a Cross-Platform and international dictionary written in Gtk2. It has powerful features such as "Glob-style pattern matching", "Scan selection word", "Fuzzy search", "Net dict", "Full text translation"; etc.	Sharepoint, .NET	Pendiente
SharpDevelop	<a href="http://www.sourceforge.net/projects/sharpdevelop/">www.sourceforge.net/projects/sharpdevelop/</a>	SharpDevelop is an open source IDE for the .NET platform. It is entirely written in C#, and supports various project types in multiple programming languages.	Sharepoint, .NET	Pendiente
SharePoint List Copy	<a href="http://www.sourceforge.net/projects/splstcp/">www.sourceforge.net/projects/splstcp/</a>	A clean and reliable way to migrate from SPS 2003 to MOSS 2007. This is a .NET/C# command line program that copies all your SharePoint 2003 lists and document libraries to a SharePoint 2007 server, keeping metadata like modification date.	Sharepoint, .NET	Pendiente
iText.NET	<a href="http://www.sourceforge.net/projects/itextdotnet/">www.sourceforge.net/projects/itextdotnet/</a>	iText.NET is iText porting on .NET Framework by J#. This library allows you to generate PDF, (X)HTML, XML, RTF files on Microsoft.NET Framework including ASP.NET.	Sharepoint, .NET	Aprobado
FCKeditor	<a href="http://www.sourceforge.net/projects/fckeditor/">www.sourceforge.net/projects/fckeditor/</a>	Online text editor (DHTML editor), for ASP, ASP.NET, ColdFusion, PHP, Java and JavaScript brings to the web many of the powerful features of known desktop editors like Word. It's XHTML compliant and works with Firefox, Mozilla, Netscape and IE.	Sharepoint, .NET	Pendiente
SharePoint Wrappers	<a href="http://www.sourceforge.net/projects/spswrappers/">www.sourceforge.net/projects/spswrappers/</a>	A set of .NET classes provides reusable objects to access Microsoft SharePoint Services via standard web service calls. These can be used by any .NET application or tool and do not require to be run on the SharePoint server.	Sharepoint, .NET	Pendiente
TripleA Maps	<a href="http://www.sourceforge.net/projects/tripleamaps/">www.sourceforge.net/projects/tripleamaps/</a>	A project by Veqryn for creating, centralizing, and hosting all open-source TripleA maps, as well as other developer resources such as artwork and map makers. If you need the actual TripleA program, go here: <a href="http://sourceforge.net/projects/triplea/">http://sourceforge.net/projects/triplea/</a>	Sharepoint, .NET	Pendiente

## Aprobación de una referencia



Lightbox control for asp

[www.sourceforge.net/projects/monit](http://www.sourceforge.net/projects/monit)

UNION OF THE SQUARES

Lightbox control for asp.net is used to display the thumbnail images, on clicking the thumbnail images

Sharepoint, .NET

Pendiente

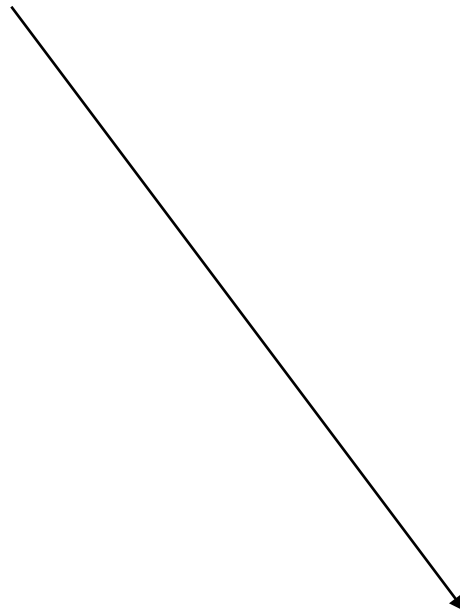
Aprobado

Pendiente

Aprobaci

# Sprint 2

Número de referencias  
por página



The screenshot shows a dialog box titled 'Enlaces Externos' with a close button (X) in the top right corner. The dialog is organized into several sections:

- Apariencia** (Appearance):
  - Título** (Title): A text field containing 'Enlaces Externos'.
  - Alto** (Height):
    - ¿Desea que el alto del elemento web sea fijo? (Do you want the web element height to be fixed?):
      - ☐ Sí (Yes): A text field followed by a 'Píxeles' (Pixels) dropdown menu.
      - ☒ No, ajustar el alto a la zona. (No, adjust the height to the area.)
  - Ancho** (Width):
    - ¿Desea que el ancho del elemento web sea fijo? (Do you want the web element width to be fixed?):
      - ☐ Sí (Yes): A text field followed by a 'Píxeles' (Pixels) dropdown menu.
      - ☒ No, ajustar el ancho a la zona. (No, adjust the width to the area.)
  - Estado de contenedor visual** (Visual container state):
    - ☐ Minimizado (Minimized)
    - ☒ Normal
  - Tipo de contenedor visual** (Visual container type): A dropdown menu set to 'Predeterminado' (Default).
- Distribución** (Distribution): A section header.
- Avanzado** (Advanced): A section header.
- Referentes Setting** (Referent Setting):
  - FilasAprobadas** (Approved rows): A text field containing '7' and a blue button with three dots.
  - FilasPendientes** (Pending rows): A text field containing '25'.

At the bottom of the dialog are three buttons: 'Aceptar' (Accept), 'Cancelar' (Cancel), and 'Aplicar' (Apply).

# Sprint 3

## Selección de Requisitos

ESTRUCTURA DE  
RESULTADOS  
CONSULTAS XML  
GENERALIZACIÓN  
Y ANÁLISIS

SPRINT 3

Nombre de elemento	Importancia
CONSULTAS XML	100
Notas	Estimación
Cómo probarlo	54h
Un sistema genera resultados, una vez se genera nada.	D

Nombre de elemento	Importancia
ESTRUCTURA DE RESULTADOS	80
Notas	Estimación
Cómo probarlo	24h
Verificar que obtenemos con cada búsqueda los campos considerados como obligatorios.	E

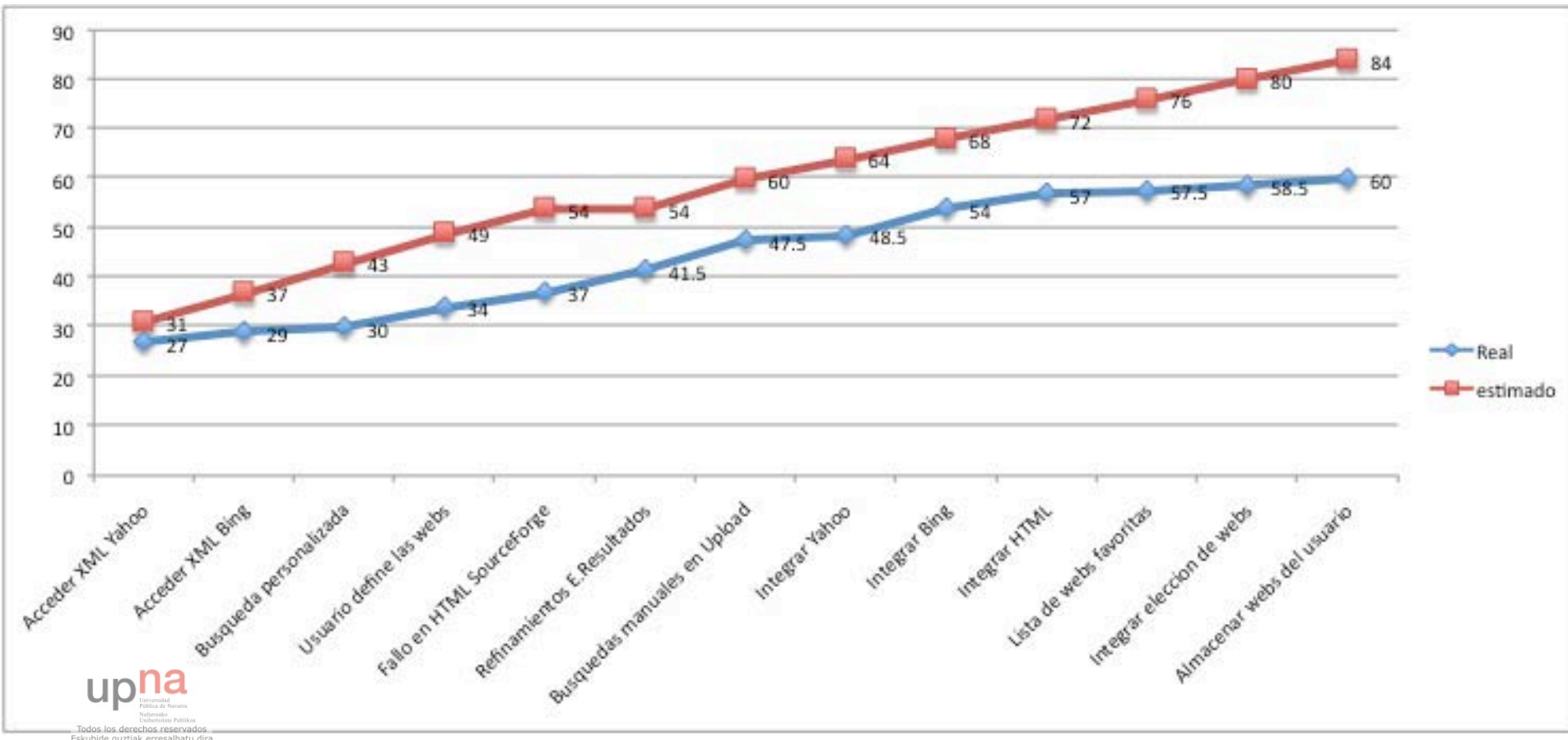
  

Nombre de elemento	Importancia
GENERALIZAR ANALISIS	30
Notas	Estimación
Cómo probarlo	30h
	F



# Sprint 3

## Seguimiento de las tareas



# Sprint 3

## Web part añadir tags

### Añadir Tags

Elija el proyecto:

Proyecto21 ▼

Añada/Edite los tags separados por comas

Xml, Html

Añadir

Marque la casilla si desea buscar referencias externas ☐

Elija un motor de búsqueda

HTML custom search ▼

Pulse el boton '>' para añadir la pagina a la búsqueda

Codeplex (HTML)  
SourceForge (HTML)



# Sprint 3

La misma funcionalidad que la web part de subida de proyectos.

## Subir Proyecto

El proyecto se ha subido correctamente. Para visitarlo pulse [aquí](#).

[Volver a subir](#)

Añada los tags separados por comas

SharePoint, 2010, C#

Añadir

Marque la casilla si desea buscar referencias externas ☐

Elija un motor de búsqueda

HTML custom search

Pulse el boton '>' para añadir la pagina a la búsqueda

Codeplex (HTML)  
SourceForge (HTML)

X

>

<

Tags añadidos correctamente

# Sprint 3

Si marcamos el *CheckBox*

Subir Proyecto

El proyecto se ha subido correctamente. Para visitarlo pulse [aquí](#).

[Volver a subir](#)

Añada los tags separados por comas

SharePoint, 2010, C#

Marque la casilla si desea buscar referencias externas ☒

Elija un motor de búsqueda

Bing

Pulse el boton '>' para añadir la pagina a la busqueda

elegir pagina..

www.codeplex.com

www.codeproject.com

X

>

<

SourceForge (HTML)

Bing: www.codeproject.com

Añadir y Buscar

# Sprint 3

Podemos elegir página..

Elija un motor de búsqueda

Bing

Pulse el boton '>' para añadir la pagina a la busqueda

elegir pagina..  
www.codeplex.com  
www.codeproject.com

X  
>  
<

Añadir y Buscar

Escriba la web que desea añadir

www.cesnavarra.net Add web

..siempre que sea válida.



## Subir Proyecto

El proyecto se ha subido correctamente. Para visitarlo pulse [aquí](#).

[Volver a subir](#)

Añada los tags separados por comas

SharePoint, C#

Marque la casilla si desea buscar referencias externas ☒

Elija un motor de búsqueda

Yahoo

Pulse el boton '>' para añadir la pagina a la búsqueda

elegir pagina.. www.msdn.com www.sourceforge.net www.codeplex.com	X > <	SourceForge (HTML) Bing: www.codeproject.com Yahoo: www.codeplex.com
Añadir y Buscar		

Buscando en HTML custom los tags: SharePoint, C#, con SourceForge (HTML)  
25 referencias encontradas  
5 referencias registradas

Buscando en www.codeproject.com los tags: SharePoint, C#, con Bing  
0 referencias encontradas  
0 referencias registradas

Buscando en www.codeplex.com los tags: SharePoint, C#, con Yahoo  
10 referencias encontradas  
10 referencias registradas

Tags añadidos correctamente

**15 referencias registradas**

# Sprint 4

## Selección de Requisitos

CORRECCIONES (!)  
AUTOMATIZACIÓN  
BUSQUEDAS  
INTEGRACIÓN  
CONSULTAS BLOGS (!)

**SPRINT 4**

Nombre de elemento	Importancia
<b>AUTOMA BUSQUEDAS</b>	90
Notas Flujo que se sigue al añadir un producto como si se añadiera ref. externa	Estimación 10h
Cómo probarlo	6

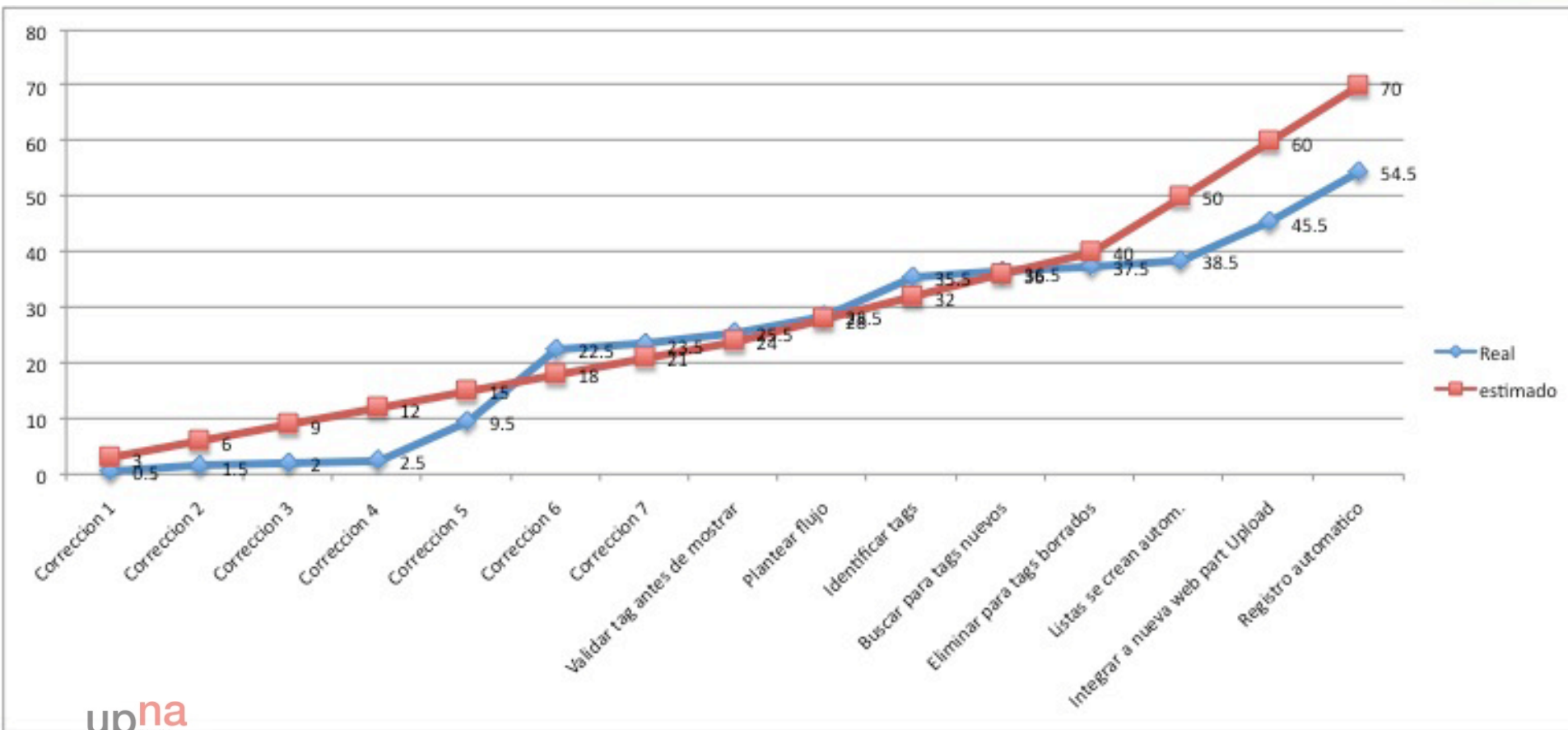
Nombre de elemento	Importancia
<b>CORRECCIONES</b>	70
Notas Correcciones propuestas a partir de la prueba de usuario	Estimación 24h
Cómo probarlo	1

Nombre de elemento	Importancia
<b>INTEGRACION</b>	90
Notas	Estimación 30h
Cómo probarlo Que funcione correctamente con la plataforma de Xebec	3

Nombre de elemento	Importancia
<b>CONSULTAS BLOG</b>	70
Notas	Estimación 24h
Cómo probarlo Realizar búsquedas en los sitios web de los blogs y que nos muestre los resultados como los que se muestran en la app	H

# Sprint 4

## Seguimiento de las tareas



# Sprint 4

## Flujo de trabajo

Actúa sobre la lista de proyectos:

Lista de Proyectos : Vista Proyecto ▾

ctos

Search this site...

<input type="checkbox"/>	ID Proyecto	Título Proyecto	Enlace al Proyecto	Estado general	Tags	Info de Tags
	1	PFC	<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC</a>	Estructura VÁLIDA	Sharepoint, .NET	
	2	Proyecto21	<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/Proyecto21">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/Proyecto21</a>	Estructura VÁLIDA	Xml, Html, web	
	3	PFC2	<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC2">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC2</a>	Estructura VÁLIDA	SharePoint, C#	

+ Agregar nuevo elemento

Si borramos una etiqueta:

<input type="checkbox"/>	ID Proyecto	Título Proyecto	Enlace al Proyecto	Estado general	Tags	Info de Tags	FlujoTags
	1	PFC	<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC</a>	Estructura VÁLIDA	Sharepoint, .NET		
	2	Proyecto21	<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/Proyecto21">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/Proyecto21</a>	Estructura VÁLIDA	Xml, Html	Tags eliminados: web	Finalizado
	3	PFC2	<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC2">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC2</a>	Estructura VÁLIDA	SharePoint, C#		



# Sprint 4

Si añadimos una etiqueta nueva:

Enlace al Proyecto	Estado general	Tags	Info de Tags	FlujoTags
<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC</a>	Estructura VÁLIDA	Sharepoint, .NET		
<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/Proyecto21">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/Proyecto21</a>	Estructura VÁLIDA	Xml, Html, language	Tags añadidos: Buscando referentes para language 0 resultados de búsqueda en www.msdn.com con Yahoo 10 resultados de búsqueda en www.codeplex.com con Bing 10 resultados de búsqueda en www.sourceforge.net con Yahoo <a href="#">language</a>	Finalizado
<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC2">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC2</a>	Estructura VÁLIDA	SharePoint, C#		

Cuando añadimos y borramos:

Enlace al Proyecto	Estado general	Tags	Info de Tags	FlujoTags
<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC</a>	Estructura VÁLIDA	Sharepoint, .NET		
<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/Proyecto21">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/Proyecto21</a>	Estructura VÁLIDA	Xml, Html, SharePoint	Tags eliminados: language (Eliminados referentes externos) Tags añadidos: <a href="#">SharePoint</a>	Finalizado
<a href="http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC2">http://win-ivgatwli52b:20294/Biblioteca%20de%20Proyectos/PFC2</a>	Estructura VÁLIDA	SharePoint, C#		



# Conclusiones y líneas futuras

## Conclusiones

- Maximización del valor
- Valoración Consultas
- Valoración Scrum

## Líneas Futuras

- Web semántica
- Integración a la plataforma

# ¿Preguntas?

